

# Dueling Posterior Sampling for Preference-Based Reinforcement Learning

Ellen R. Novoseller<sup>1</sup>, Yanan Sui<sup>2</sup>, Yisong Yue<sup>1</sup>, Joel W. Burdick<sup>1</sup>

<sup>1</sup> Department of Computing and Mathematical Sciences, California Institute of Technology  
{enovoseller@caltech.edu, yyue@caltech.edu, jwb@robotics.caltech.edu}

<sup>2</sup> Department of Computer Science, Stanford University, ysui@cs.stanford.edu

## Abstract

In preference-based reinforcement learning (RL), an agent interacts with the environment while receiving preferences instead of absolute feedback. While there is increasing research activity in preference-based RL, the design of formal frameworks that admit tractable theoretical analysis remains an open challenge. Building upon ideas from preference-based bandit learning and posterior sampling in RL, we present DUELING POSTERIOR SAMPLING (DPS), which employs preference-based posterior sampling to learn both the system dynamics and the underlying utility function that governs the user’s preferences. Because preference feedback is provided on trajectories rather than individual state/action pairs, we develop a Bayesian approach to solving the credit assignment problem, translating user preferences to a posterior distribution over state/action reward models. We prove an asymptotic no-regret rate for DPS with a Bayesian logistic regression credit assignment model; to our knowledge, this is the first regret guarantee for preference-based RL. We also discuss possible avenues for extending this proof methodology to analyze other credit assignment models. Finally, we evaluate the approach empirically, showing competitive performance against existing baselines.

## 1 Introduction

In many domains, ranging from clinical trials [40] to autonomous driving [36] and human-robot interaction [26], it can be unclear how to define a reward signal for reinforcement learning (RL). In such situations, the RL agent seeks to interact optimally with a human user; thus, rewards should reflect the extent to which the algorithm achieves the user’s goals. Yet, for many systems, for instance in autonomous driving [8] and robotics [7, 3], users have difficulty with both specifying numerical reward functions and providing demonstrations of desired behavior. Furthermore, a misspecified reward function can result in “reward hacking” [6], which occurs when the agent learns an undesirable behavior that through some loophole, achieves a high reward. In such cases, the user’s *preferences* form a more reliable measure of desired system behavior, and the preference data may be leveraged in place of a standard numerical reward signal.

We thus study the problem of preference-based reinforcement learning (PBRL), where the RL agent executes a pair of trajectories, and the user provides (noisy) preference feedback regarding which trajectory has higher utility. While the study of PBRL has seen increased interest in recent years [18, 13, 48], it remains an open challenge to design formal frameworks that admit tractable theoretical analysis. Compared to the preference-based bandit setting, which has seen significant theoretical progress (e.g., [53, 56, 2, 44, 15, 55, 32, 52, 41, 42]), one major challenge is how to address credit assignment when only receiving feedback at the trajectory level compared to the state/action level.

In this paper, we present DUELING POSTERIOR SAMPLING (DPS), which uses preference-based posterior sampling to tackle the PBRL problem in the Bayesian regime. Posterior sampling (also

known as Thompson sampling) [45, 29, 20, 1, 30] is a Bayesian model-based approach to balancing exploration and exploitation, thereby enabling the algorithm to efficiently learn models of both the environment’s state transition dynamics and the reward function. Previous work on posterior sampling in RL [29, 20, 1, 30] all focused on learning from absolute rewards, while we show how to extend posterior sampling to both elicit and learn from trajectory-level preference feedback.

To elicit preference feedback, at every episode of learning, DPS draws two independent samples from the posterior to generate two trajectories. This approach is inspired by the Self-Sparring algorithm proposed for the bandit setting [41]. Our theoretical analysis is quite different from that in [41], due to the need to incorporate trajectory-level preference learning and state transition dynamics.

To learn from preference feedback, DPS internally maintains a Bayesian state/action reward model that explains the preferences. In other words, this reward model is a solution to the *temporal credit assignment problem* [3, 56, 44, 13, 51, 48] and determines which of the encountered states and actions are responsible for the trajectory-level preference feedback. Learning from trajectory-level preferences is in general a very challenging problem, as information about the rewards is sparse (often just one bit), is only relative to the pair of trajectories being compared, and does not explicitly include information about actions within trajectories. We thus develop our approach while restricting to standard Bayesian realizability assumptions inherent to most posterior sampling approaches.

We developed DPS concurrently with an analysis framework for characterizing regret convergence in the episodic learning setting. To justify our overall approach, we show how to mathematically integrate Bayesian credit assignment and draw dueling samples within the conventional posterior sampling framework. We evaluate several possible Bayesian credit assignment models, and prove an asymptotic no-regret rate for DPS using Bayesian logistic regression [5, 28] as the credit assignment model. To our knowledge, this is the first PBRL approach with theoretical guarantees. In addition, we also demonstrate that DPS delivers competitive performance in simulation.

## 2 Related work

**Posterior sampling.** Balancing exploration and exploitation is a key problem in reinforcement learning (RL) and bandits. In the episodic learning setting, the agent typically aims to balance exploration and exploitation to minimize its regret, i.e., the gap between the expected total rewards of the agent and the optimal policy. Posterior sampling, first proposed in [45], is a Bayesian approach toward achieving this goal, and iterates between (1) updating the posterior of a Bayesian environment model and (2) sampling from this posterior to inform the subsequent policy. In both the bandit and RL settings, posterior sampling has been demonstrated to perform competitively in experiments and enjoy favorable theoretical properties in terms of its regret [30, 29, 1, 12].

Our approach builds upon two prior posterior sampling algorithms: Self-Sparring [41] for preference-based bandit learning (also known as dueling bandits [53]) and posterior sampling RL [29]. Self-Sparring [41] is a posterior sampling approach, and draws multiple samples to “duel” or “spar” via preference elicitation. The algorithm iteratively: a) draws multiple samples from the posterior model of each action’s reward; b) for each sampled model, executes the action with the highest sampled reward; c) queries for preference feedback between the executed actions; and d) updates the posterior according to the acquired preference data. In [41], the authors prove an asymptotic no-regret guarantee for Self-Sparring with independent Beta-Bernoulli reward models for each action.

Within RL, posterior sampling has been applied to the finite-horizon setting with absolute rewards [29]. Posterior sampling RL iterates over four steps: a) draw a sample from the Bayesian posterior of the dynamics and rewards; b) compute the optimal policy for the sampled system; c) execute the policy to get a roll-out trajectory; and d) update the posteriors with the new observations from the roll-out. In [29], the authors show the expected regret is  $O(hS\sqrt{AT\log(SAT)})$ , for number of time-steps  $T$ , finite time horizon  $h$ , and discrete state and action spaces of sizes  $S$  and  $A$ , respectively.

A third line of relevant work is posterior sampling for Bayesian logistic regression [12, 16, 37], which is used as our Bayesian credit assignment model. One difficulty with Bayesian logistic regression [5] is the lack of a closed-form posterior. To handle this, we adopt the approach of [12] and use a Laplace approximation. Other approaches include using Gibbs sampling algorithm [16]. One relevant related application is [35], who apply Bayesian logistic regression to the multi-objective multi-armed bandit

problem; to determine the utilities that a human assigns to different objectives, the algorithm queries for pairwise preferences between expected reward vectors corresponding to different actions.

**Preference-based learning.** Previous work on preference-based RL (PBRL) has shown successful performance in a number of applications, such as playing games [13], learning human preferences for autonomous driving [36], and selecting a robot’s controller parameters [26, 4]. Yet, to our knowledge, the PBRL literature still lacks theoretical guarantees.

Existing approaches for trajectory-level preference-based RL may be broadly divided into three categories [47]: a) directly optimizing policy parameters [46, 11, 26]; b) learning a preference model to predict action preferences in each state [18]; and c) learning a utility function to characterize the rewards, returns, or values of state/action pairs [49, 50, 3, 51, 13]. In c), the utility is often modeled as linear in the trajectory features. If those features are defined as visitations to each state/action pair, then maximizing utility directly corresponds to maximizing the total (undiscounted) reward.

One popular paradigm, which we also adopt, is PBRL with underlying utility functions. By inferring state/action rewards from preference feedback, one can derive relatively-interpretable reward models and also use such methods as value iteration. In addition, utility-based approaches may be more sample efficient compared to policy search and preference relation methods [47], as they extract more information from each observation. Notably, [46] learn a Bayesian model over policy parameters, and draw samples from its posterior to inform actions. From existing PBRL methods, their algorithm perhaps most resembles ours; however, compared to utility-based approaches, policy search methods typically require either more samples or expert knowledge to craft the policy parameters [48, 26].

Beyond RL, preference-based learning has been the subject of much research. The closest to RL is the bandit setting [53, 56, 2, 44, 15, 55, 32, 52, 41, 42], which is essentially a single-state variant of RL. Other settings include: active learning [36, 23, 17], which is focused exclusively on learning an accurate model rather than maximizing utility of decision-making; learning with more structured preference feedback [31, 38, 33, 39], where the learner receives more than one bit of information per preference elicitation; and batch supervised settings such as learning to rank [22, 14, 25, 9, 54, 10, 27].

### 3 Problem statement

**Preliminaries.** We consider fixed-horizon Markov Decision Processes (MDPs), in which rewards are replaced by preferences over trajectories. This class of MDPs can be represented as a tuple,  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \succ, \phi, p, p_0, h)$ , where the state space  $\mathcal{S}$  and action space  $\mathcal{A}$  are finite sets. The agent, using policy  $\pi$ , episodically interacts with the environment with length- $h$  roll-out trajectories of the form  $\tau = \{s_1, a_1, s_2, a_2, \dots, s_h, a_h, s_{h+1}\}$ . Since we are eliciting preference feedback, in each episode  $i$ , the agent executes two roll-outs  $\tau_{i1}$  and  $\tau_{i2}$ , and observes a preference between the two. The initial state is sampled from  $p_0$ , while  $p$  defines the transition dynamics:  $s_{t+1} \sim p(\cdot | s_t, a_t)$ .

We use  $\succ$  to denote the stochastic preference relationship between trajectories, and  $\phi(\tau, \tau') = \mathbb{P}(\tau \succ \tau') \in [0, 1]$  to capture the feedback generation mechanism. We assume that  $\succ$  is a total ordering over trajectories, and  $\tau \succ \tau' \Leftrightarrow \phi(\tau, \tau') > \frac{1}{2}$ . We use  $\tau > \tau'$  to denote the event that trajectory  $\tau$  was preferred over  $\tau'$  in a preference elicitation, i.e.,  $\tau \succ \tau'$  is observed with probability  $\phi(\tau, \tau')$ . We further assume an underlying utility function  $\bar{r}(\tau)$  for each trajectory, such that  $\tau \succ \tau' \Leftrightarrow \bar{r}(\tau) > \bar{r}(\tau')$ , and define  $\phi$  using  $\bar{r}$ . For instance, if the preferences are noiseless, then:  $\phi(\tau_i, \tau_j) = \mathbb{I}[\bar{r}(\tau_i) > \bar{r}(\tau_j)]$ . Alternatively,  $\phi$  could be the linear link function [2]:  $\phi_{\text{lin}}(\tau_i, \tau_j) := (1 + \bar{r}(\tau_i) - \bar{r}(\tau_j))/2$ . We primarily assume a logistic or Bradley-Terry link function:  $\phi_{\text{log}}(\tau_i, \tau_j) := [1 + \exp(-c(\bar{r}(\tau_i) - \bar{r}(\tau_j)))]^{-1}$  with “temperature”  $c \in (0, \infty)$ . Our problem setting resembles the PSDP defined in [50], except that additionally, we incorporate the noise model through which the underlying utilities are stochastically translated to preferences. Finally, we assume that the utilities decompose additively:  $\bar{r}(\tau) \equiv \sum_{i=1}^h \bar{r}(s_i, a_i)$  for state/action pairs in  $\tau$ .

Given a policy  $\pi$ , we can define the standard RL value function as the expected total utility of being in state  $s$  at step  $i$ , and following policy  $\pi$ :

$$V_{\pi,i}(s) = \mathbb{E} \left[ \sum_{j=i}^h \bar{r}(s_j, \pi(s_j)) | s_i = s \right], \quad (1)$$

---

**Algorithm 1** DUELING POSTERIOR SAMPLING (DPS)

---

```
 $H = \emptyset$  {Initialize history}  
 $T = \emptyset$  {Initialize list of preference data}  
Initialize prior for  $f$  {Initialize state transition model}  
Initialize prior for  $g$  {Initialize utility model}  
while True do  
   $\pi_1 \leftarrow \text{ADVANCE}(H, T, f, g)$   
   $\pi_2 \leftarrow \text{ADVANCE}(H, T, f, g)$   
  Sample trajectories  $\tau_1$  and  $\tau_2$  from  $\pi_1$  and  $\pi_2$   
  Observe feedback  $b = \mathbb{I}(\tau_2 > \tau_1)$   
   $H = H \cup (s_1^{\tau_1}, a_1^{\tau_1}, s_2^{\tau_1}) \cup \dots \cup (s_h^{\tau_2}, a_h^{\tau_2}, s_{h+1}^{\tau_2})$   
   $T = T \cup (\tau_1, \tau_2, b)$   
   $\text{FEEDBACK}(H, T, f, g)$   
end while
```

---

and now we can define the optimal policy  $\pi^*$  as the one with maximal value for all input states. Note that  $\mathbb{E}_{s_1 \sim p_0} [V_{\pi,1}(s_1)] \equiv \mathbb{E}_{\tau \sim \pi, \mathcal{M}} [\bar{r}(\tau)]$ . Given fully specified dynamics and reward models,  $p$  and  $\bar{r}$ , it is straightforward to apply standard dynamic programming approaches such as value iteration to arrive at the optimal policy under  $p$  and  $\bar{r}$  [43]. The goal of learning, then, is infer  $p$  and  $\bar{r}$  to the extent necessary for good decision-making.

**Learning problem.** In each iteration (or episode)  $i$ , the agent selects two policies,  $\pi_{i1}$  and  $\pi_{i2}$ . The two policies are rolled out to obtain trajectories  $\tau_{i1}$  and  $\tau_{i2}$ , and a binary preference  $b_i \in \{0, 1\}$  between them is sampled according to the underlying utilities of  $\tau_{i1}$  and  $\tau_{i2}$ . We quantify the performance of the learning agent using expected cumulative regret relative to the optimal policy:

$$\mathbb{E}[\text{REG}_T] = \mathbb{E} \left\{ \sum_{i=1}^{\lceil T/(2h) \rceil} \sum_{s \in \mathcal{S}} p_0(s) [2V_{\pi^*,1}(s) - V_{\pi_{i1},1}(s) - V_{\pi_{i2},1}(s)] \right\}. \quad (2)$$

To minimize regret, the agent must balance exploration (collecting new data) with exploitation (behaving optimally w.r.t. existing models). Over-exploration of bad trajectories will incur large regret, and under-exploration can prevent converging to the optimal policy. In contrast to the standard formulation in RL [29], at each iteration/episode we compare the utilities of both selected policies.

## 4 Algorithm

As outlined in Algorithm 1, DUELING POSTERIOR SAMPLING (DPS) iterates over three main steps: (a) sample two policies  $\pi_1, \pi_2$  from the Bayesian posteriors of the dynamics and utility models (ADVANCE – Algorithm 2); (b) roll out  $\pi_1$  and  $\pi_2$  to obtain trajectories  $\tau_1$  and  $\tau_2$ , and receive preference feedback between them; (c) store the new state transitions and feedback and update the posterior (FEEDBACK – Algorithm 3). Compared to conventional posterior sampling with absolute feedback [29], the two key differences are that: two policies are sampled rather than one each iteration, and a credit assignment problem is solved when learning from feedback.

ADVANCE (Algorithm 2) samples from the Bayesian posteriors of the dynamics and utility models. The sampled dynamics and utilities form an MDP, and value iteration is used to derive the optimal policy  $\pi$  under the sample. One can also think of  $\pi$  as a random function whose randomness depends on the sampling of the dynamics and utility models. In the Bayesian setting, it can be shown that  $\pi$  is sampled according to its posterior probability of being the true optimal policy  $\pi^*$  [29, 30]. Intuitively, peaked (i.e., certain) posteriors lead to less variability when sampling  $\pi$ , which implies less exploration. On the other hand, diffuse (i.e., uncertain) posteriors lead to greater variability when sampling  $\pi$ , which implies more exploration.

FEEDBACK (Algorithm 3) updates the Bayesian posteriors of the dynamics and utility models based on new data. Updating the dynamics posterior is relatively straightforward, as we assume that the dynamics are fully-observed; for instance, the dynamics prior can be modeled via Dirichlet distributions with multinomial conjugate observation likelihoods [29]. In contrast, performing Bayesian inference over state/action utilities from trajectory-level feedback is much more challenging. We considered a range of approaches (see Appendix A1), and found Bayesian logistic regression (Section 4.1) to both be well-performing and admit tractable analysis within our theoretical framework.

---

**Algorithm 2** ADVANCE: Sample policy from dynamics and utility models

---

**Input:**  $H, T, f, g$   
Sample  $M \sim f(\cdot|H)$  {Sample MDP transition dynamics from posterior}  
Sample  $R \sim g(\cdot|T)$  {Sample utilities from posterior}  
Compute  $\pi = \operatorname{argmax}_{\pi} V(M, R)$  {Value iteration yields sampled MDP's optimal policy}  
Return  $\pi$

---

---

**Algorithm 3** FEEDBACK: Update dynamics and utility models based on new user feedback

---

**Input:**  $H, T, f, g$   
Apply Bayesian update to  $f$ , given  $H$  {Update dynamics model given history}  
Apply Bayesian update to  $g$ , given  $T$  {Update utility model given preferences}  
Return  $f, g$

---

#### 4.1 Bayesian logistic regression for utility inference and credit assignment

*Credit assignment* [47] is the problem of inferring which state/action pairs are responsible for observed trajectory-level preferences. We detail a Bayesian logistic regression approach to address this task in our setting. Logistic regression is a binary classification method that learns a weight vector  $\mathbf{w}$  for the model  $p(y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$ . Bayesian logistic regression [5, 28] maintains a posterior over possible weight vectors. Because there is no convenient prior yielding a closed-form conjugate posterior, we use the Laplace approximation to the posterior as specified below.

**Preliminaries.** Let  $N$  be the number of trajectories pairs observed so far, and  $D = SA$  be the total number of state/action pairs. Let  $\mathbf{x}_{ij} \in \mathbb{R}^D, j \in \{1, 2\}$  be the visitation vector corresponding to trajectory  $\tau_{ij}$ , with the  $k^{\text{th}}$  element  $x_{ij}^{(k)}$  being the number of times that state/action pair  $k$  was visited in  $\tau_{ij}$ . Define  $\mathbf{x}_i := \mathbf{x}_{i1} - \mathbf{x}_{i2}$ . The observation matrix  $X$  and label vector  $\mathbf{y}$  are defined as:

$$X = \begin{bmatrix} (\mathbf{x}_{11} - \mathbf{x}_{12})^T \\ \vdots \\ (\mathbf{x}_{N1} - \mathbf{x}_{N2})^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 2\mathbb{I}_{[\tau_{11} > \tau_{12}]} - 1 \\ \vdots \\ 2\mathbb{I}_{[\tau_{N1} > \tau_{N2}]} - 1 \end{bmatrix}, \quad (3)$$

where the expression  $2\mathbb{I}_{[\tau_{i1} > \tau_{i2}]} - 1$  results in labels  $y_i$  with values in  $\{-1, 1\}$ .

The observation matrix  $X \in \mathbb{R}^{N \times D}$  has rank at most  $D-1$ , since the elements of  $\mathbf{x}_i = \mathbf{x}_{i1} - \mathbf{x}_{i2}$  must sum to zero for each row. To obtain a full-row-rank observation matrix for Bayesian logistic regression, we transform  $X \in \mathbb{R}^{N \times D}$  to  $Z \in \mathbb{R}^{N \times (D-1)}$  via the matrix  $V = [\mathbf{v}_1 \ \dots \ \mathbf{v}_{D-1}] \in \mathbb{R}^{D \times (D-1)}$ , where the columns  $\mathbf{v}_i \in \mathbb{R}^D$  form an orthonormal basis spanning the  $(D-1)$ -dimensional, full possible row space of  $X$ . To obtain the vector  $\mathbf{z}_i \in \mathbb{R}^{D-1}$  that expresses  $\mathbf{x}_i$  in this basis, apply:

$$\mathbf{z}_i = [\mathbf{x}_i^T \mathbf{v}_1 \ \dots \ \mathbf{x}_i^T \mathbf{v}_{D-1}]^T = V^T \mathbf{x}_i, \quad (4)$$

while converting any vector  $\mathbf{z}_i \in \mathbb{R}^{D-1}$  back to the original basis can be accomplished via:

$$\mathbf{x}_i = \sum_{j=1}^{D-1} z_{ij} \mathbf{v}_j = V \mathbf{z}_i, \text{ where } z_{ij} \text{ is the } j^{\text{th}} \text{ element of } \mathbf{z}_i. \quad (5)$$

Note that this transformation preserves inner products. Equation (5) can be applied to show:

$$\mathbf{x}_1^T \mathbf{x}_2 = \left( \sum_{i=1}^{D-1} z_{1i} \mathbf{v}_i \right)^T \left( \sum_{j=1}^{D-1} z_{2j} \mathbf{v}_j \right) = \mathbf{z}_1^T \mathbf{z}_2, \text{ by orthonormality of } \{\mathbf{v}_i\}. \quad (6)$$

In particular, the transformation preserves orthogonality, so that  $X$  and  $Z$  have the same row-rank and  $X^T X$  and  $Z^T Z$  have the same rank.

**Utility model & posterior inference.** We fit a Bayesian logistic regression model to the transformed data  $(Z, \mathbf{y})$ . Afterwards, this model predicts the probability that  $\tau$  is preferred to  $\tau'$  as a logistic regression function of their visitation vector differences  $\mathbf{x}_{\tau} - \mathbf{x}_{\tau'}$ . The model parameters correspond exactly to the state/action utilities  $\bar{r}$ . The model internally computes an element-wise product between  $\mathbf{x}_{\tau} - \mathbf{x}_{\tau'}$  and estimated reward vector  $\bar{r}$ , within the  $(D-1)$ -dimensional space given by (4). Given

the inner product equivalence (6), this is exactly the trajectory utility, and taking the expectation over trajectories generated by a policy is exactly the value function (1). We show in our experiments that Bayesian logistic regression can robustly learn even with preference modeling mismatch.

We are chiefly interested in sampling from the posterior of parameter/utility vector  $\bar{\mathbf{r}} \in \mathbb{R}^D$ , which can be combined with the sampled dynamics to perform value iteration and obtain a policy. As shown below, via the Laplace approximation, the posterior is Gaussian distributed, and thus can be easily sampled. The internal utility representation lies in  $\tilde{\mathbf{r}}' \in \mathbb{R}^{D-1}$ , and we convert to  $\tilde{\mathbf{r}} \in \mathbb{R}^D$  via (5).

We now describe the Bayesian logistic regression step itself. A Gaussian prior is defined over the utilities  $\mathbf{r}' \in \mathbb{R}^{D-1}$ :  $p(\mathbf{r}') \sim \mathcal{N}(\mathbf{r}' | \mathbf{r}'_0, V'_0)$ . The logistic regression likelihood is:

$$p(Z, \mathbf{y} | \mathbf{r}') = \prod_{i=1}^N p(z_i, y_i | \mathbf{r}') = \prod_{i=1}^N \frac{1}{1 + \exp(-y_i \mathbf{z}_i^T \mathbf{r}')} \quad (7)$$

We approximate the posterior as Gaussian via the Laplace approximation:

$$p(\mathbf{r}' | Z, \mathbf{y}) \approx \mathcal{N}(\mathbf{r}' | \hat{\mathbf{r}}', H^{-1}), \text{ where:} \quad (8)$$

$$\hat{\mathbf{r}}' = \underset{\mathbf{r}'}{\operatorname{argmin}} f(\mathbf{r}'), \quad f(\mathbf{r}') := -\log p(Z, \mathbf{y}, \mathbf{r}') = -\log p(\mathbf{r}') - \log p(Z, \mathbf{y} | \mathbf{r}'), \quad (9)$$

$$H = \nabla_{\mathbf{r}'}^2 f(\mathbf{r}') \Big|_{\hat{\mathbf{r}}'}, \text{ and where the optimization problem in (9) is convex.} \quad (10)$$

To show a regret convergence using this approximate posterior, we leverage asymptotic normality of the maximum likelihood estimator of logistic regression in our proofs.

## 5 Theoretical results

We now sketch our asymptotic no-regret analysis for DUELING POSTERIOR SAMPLING (DPS) with Bayesian logistic regression. The full proof is in Appendix A2, while Appendix A2.1 discusses possible avenues for extending this proof methodology toward additional credit assignment models. The proof has two main parts: first proving that DPS with logistic credit assignment is asymptotically consistent (Theorem 1), and then proving that DPS has a sublinear regret rate (Theorem 2). Both parts leverage results on the asymptotic behavior of logistic regression [21]. As before, we consider data  $Z \in \mathbb{R}^{N \times (D-1)}$  and labels  $\mathbf{y} \in \mathbb{R}^N$ , with  $[Z]_{ij} = z_{ij}$ . To show that DPS is asymptotically consistent in learning the reward function, we first provide some definitions and necessary conditions.

**Definition 1** (Derivative of sigmoid).  $f : \mathbb{R} \rightarrow \mathbb{R}$ , where  $f(x) = \frac{e^{-x}}{(1+e^{-x})^2}$ . Note  $f(x) = f(-x)$ .

**Definition 2.** Let  $\bar{\mathbf{r}} \in \mathbb{R}^D$  be the vector of true state/action utilities (assumed to exist) and  $\bar{\mathbf{r}}' \in \mathbb{R}^{D-1}$  be its transformation via (4). Define  $\tilde{\mathbf{r}}'_k \in \mathbb{R}^{D-1}$  as the state/action rewards sampled from the Bayesian logistic regression model posterior in episode  $k$ ,  $\hat{\mathbf{r}}'_k \in \mathbb{R}^{D-1}$  as the model's maximum a posteriori (MAP) estimate at episode  $k$ , and  $\hat{\mathbf{r}}'_{ML,k} \in \mathbb{R}^{D-1}$  as its maximum likelihood estimate at  $k$ . Lastly,  $\tilde{\mathbf{r}}_k \in \mathbb{R}^D$ ,  $\hat{\mathbf{r}}_k \in \mathbb{R}^D$ , and  $\hat{\mathbf{r}}_{ML,k} \in \mathbb{R}^D$  are their respective equivalents given by (5).

**Condition 1.**  $\exists m_0 < \infty$  such that  $|z_{ij}| \leq m_0$  for all  $i \in \{1, \dots, N\}, j \in \{1, \dots, D-1\}$ .

**Condition 2.** Let  $\lambda_1^{(k)}$  and  $\lambda_{D-1}^{(k)}$  be the largest and smallest eigenvalues, respectively, of  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$ . Then,  $\exists m_1 < \infty$  such that  $\frac{\lambda_1^{(k)}}{\lambda_{D-1}^{(k)}} < m_1$ , for all  $k$ .

**Proposition 1** (Asymptotic consistency of logistic regression [21]). *If Conditions 1 and 2 are satisfied, then the maximum likelihood estimator  $\hat{\mathbf{r}}'_{ML,k}$  of  $\bar{\mathbf{r}}'$  exists almost surely as  $k \rightarrow \infty$ , and  $\hat{\mathbf{r}}'_{ML,k}$  converges almost surely to the true values  $\bar{\mathbf{r}}'$  if and only if  $\lim_{k \rightarrow \infty} \lambda_{D-1}^{(k)} = \infty$ .*

We first show that Proposition 1's final condition is satisfied with known transition dynamics, and afterwards consider the convergence of the dynamics model posterior.

**Lemma 1.** *Under known transition dynamics, all eigenvalues of the matrix  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$  approach infinity as  $k \rightarrow \infty$ .*

**Lemma 2** (Convergence of dynamics model). *Given Lemma 1, DPS's dynamics model converges to the true dynamics, and as it converges, all eigenvalues of  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$  approach infinity.*

Combining these results, we obtain:

**Theorem 1** (Asymptotic consistency of DPS). *If there exists a reward function such that a logistic regression model explains the preference feedback, then DPS with a Bayesian logistic regression credit assignment model will learn an asymptotically consistent reward model.*

We turn next to characterizing the regret rate of DPS. We apply two prior results, one from [21] regarding the asymptotic distribution of the logistic regression maximum likelihood estimate (Prop. 2), and the other from [29] regarding a regret bound for posterior sampling RL (Prop. 3).

**Proposition 2** (Asymptotic normality of logistic regression maximum likelihood estimator [21]). *If Conditions 1 and 2 are satisfied, and if  $\hat{\mathbf{r}}'_{ML,k}$  converges almost surely to the true value  $\bar{\mathbf{r}}'$ , then:*

$$\left[ \sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_{ML,k}) \mathbf{z}_i \mathbf{z}_i^T \right]^{\frac{1}{2}} (\hat{\mathbf{r}}'_{ML,k} - \bar{\mathbf{r}}') \xrightarrow{D} \mathcal{N}(\mathbf{0}, \mathbb{I}) \text{ as } k \rightarrow \infty, \quad (11)$$

where  $\xrightarrow{D}$  implies convergence in distribution and  $Q^{\frac{1}{2}}$  is the positive definite matrix associated with positive definite matrix  $Q$  such that  $[Q^{\frac{1}{2}}]^2 = Q$ .

**Proposition 3** (Expected regret of posterior sampling RL [29]). *Posterior sampling RL has expected  $T$ -step regret  $O(hS\sqrt{AT\log(SAT)})$ , with horizon  $h$  and numbers of states and actions  $S$  and  $A$ .*

Leveraging these results, we show that under preference feedback, the regret can be decomposed into two terms: one that reflects the converging dynamics model, and one that reflects the converging reward model (inferred from trajectory-level preference feedback).

**Lemma 3** (Regret decomposition). *The expected regret of DPS can be decomposed into two terms. One term can be bounded by the regret bound of [29], stated in Proposition (3). The other is bounded by:  $h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{\mathbf{r}} - \tilde{\mathbf{r}}_k\|_\infty] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\hat{\mathbf{r}}_k - \bar{\mathbf{r}}\|_\infty] + h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\hat{\mathbf{r}}_k - \tilde{\mathbf{r}}_k\|_\infty]$ .*

The final result is obtained by analyzing convergence of the samples  $\tilde{\mathbf{r}}_k$  to  $\hat{\mathbf{r}}_k$  and of the credit assignment model  $\hat{\mathbf{r}}_k$  to  $\bar{\mathbf{r}}$ :

**Theorem 2** (Asymptotic regret rate of DPS). *If there exists a reward function such that a logistic regression model explains the preferences, then DPS has an asymptotic no-regret rate of  $O\left(hS\sqrt{AT\log(SAT)} + h\sqrt{\frac{SA}{c_0}T\log(T)}\right)$ , where  $c_0$  is a minimum rate at which eigenvalues of  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$  increase linearly with collection of samples  $\mathbf{z}_i$  that impact those eigenvalues.*

## 6 Experiments

We validate the empirical performance of DUELING POSTERIOR SAMPLING (DPS) on two simulated domains with varying levels of preference noise, as well as using alternative credit assignment models. We find that DPS generally performs well and outperforms standard PBRL baselines [49].

**Experimental setup.** We evaluate on two simulated environments: RiverSwim and random MDPs. The RiverSwim environment [29] has six states and two actions (actions 0 and 1); the optimal policy is to always choose action 1, which maximizes the probability of reaching a particular goal state/action pair. Meanwhile, a suboptimal policy—yielding a much smaller reward compared to the goal—is quickly and easily discovered and incentivizes the agent to always select action 0. The algorithm must demonstrate sufficient exploration to have hope of discovering the optimal policy quickly.

In the second simulated environment, we generate random MDPs as in [29] with 10 states and 5 actions. The transition dynamics and rewards are generated from Dirichlet (all parameters set to 0.1) and exponential (rate parameter set to 5) distributions, respectively. The parameter for these distributions were chosen to generate MDPs with sparse dynamics and rewards. The sampled reward vectors were shifted and normalized so that the minimum reward is zero and their mean is one.

In both of these environments, preferences between pairs of trajectories were generated by (noisily) comparing the total rewards that they accumulated; this reward information was hidden from the learning algorithm, which observed only the trajectory preferences and state transitions. Preference noise is generated according to a logistic model: for trajectories  $\tau_i$  and  $\tau_j$ ,  $P(\tau_i > \tau_j) = \{1 +$

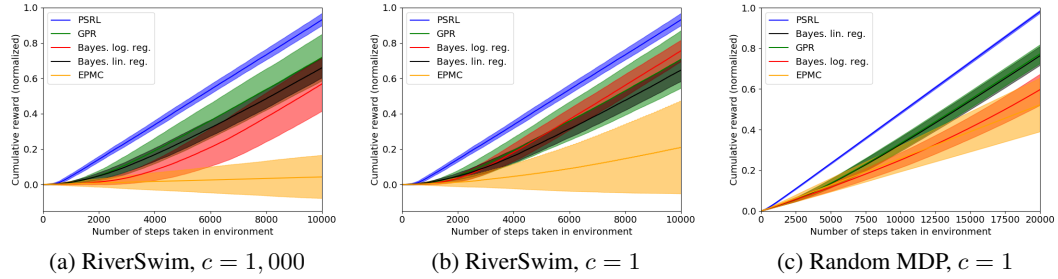


Figure 1: Empirical performance of DPS. a) and b) show RiverSwim with noise hyperparameters  $c = 1,000, 1$ . c) displays random MDPs with  $c = 1$ . Posterior sampling RL (PSRL) [29] is an upper bound that receives numerical rewards; Gaussian process regression (GPR), Bayesian linear regression, and Bayesian logistic regression are all instances of DPS. EPMC is a baseline from [50] as discussed. Plots display mean  $\pm$  one std over 100 runs of each algorithm tested. Additional results (more values of  $c$ ) are in Appendix A3. Normalization is with respect to the total reward achieved by the optimal policy. Overall, we see that DPS performs well and is robust to the choice of credit assignment model.

$\exp[-c(\bar{r}(\tau_i) - \bar{r}(\tau_j))]^{-1}$ , where  $\bar{r}(\tau_i)$  and  $\bar{r}(\tau_j)$  are the total rewards accrued by the two trajectories, respectively, while the hyperparameter  $c$  controls the degree of noisiness.

**Methods compared.** We evaluate DPS under four different noise levels ( $c \in \{0.1, 0.5, 1, 1000\}$ ) and three credit assignment models: 1) Bayesian logistic regression, 2) Bayesian linear regression, and 3) Gaussian process regression, where the latter two methods are described in Appendix A1. In addition, we evaluate the Every-Visit Preference Monte Carlo (EPMC) algorithm with probabilistic credit assignment [50, 47] as a baseline. Lastly, we compare against the posterior sampling RL algorithm [29], which learns using the true numerical rewards at each step, and thus yields an upper-bound on the performance that a preference-based algorithm could achieve.

**Results.** Figure 1 shows the performance comparison for  $c = 1$  in both environments, as well as  $c = 1,000$  in RiverSwim (additional results are in Appendix A3, including hyperparameter details). DPS performs well in all simulations, and significantly outperforms the EPMC baseline. This may be because the EPMC algorithm uses a uniform exploration strategy, while DPS prioritizes exploration by sampling high rewards in more uncertain regions of the state/action space. Notice that  $c = 1,000$  results in nearly-noiseless preferences; this can decrease performance in RiverSwim in some cases, since preference noise can help the agent to escape the local minimum. We also see that DPS is competitive with PSRL, which has access to the full cardinal rewards at each state/action. Finally, we see that the performance of DPS is robust to the choice of credit assignment model, and in fact using Gaussian process regression (for which we do not have an end-to-end regret analysis) often leads to the best empirical performance. These results suggest that DPS is a practically promising approach that can robustly incorporate many modeling approaches as subroutines.

## 7 Conclusion

We investigate the preference-based reinforcement learning problem, which receives comparative preferences instead of absolute real-valued rewards as feedback. We develop the DUELING POSTERIOR SAMPLING (DPS) algorithm, which optimizes policies in an highly efficient and flexible way. To our knowledge, DPS is the first preference-based RL algorithm with a regret guarantee. DPS also performs well in our simulations, and seems practically promising. That makes it both a theoretically-justified and practically promising algorithm.

There are many directions for future work. The Bayesian logistic regression model could be improved with more accurate posterior estimates. Assumptions governing the user’s preferences, such as requiring an underlying utility model, could be relaxed. One can also incorporate kernelized methods to further improve sample efficiency. It is also important to extend to other credit assignment models, such as the Gaussian process regression and Bayesian linear regression methods, for which the same concept of the regret decomposition still applies. We expect that DPS would perform well with any asymptotically consistent reward model that sufficiently captures users’ preference behavior.



## Acknowledgments

This work was supported by NIH grant EB007615 and an Amazon graduate fellowship.

## References

### References

- [1] Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: Worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pages 1184–1194, 2017.
- [2] Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pages 856–864, 2014.
- [3] Riad Akrou, Marc Schoenauer, and Michèle Sebag. APRIL: Active preference learning-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 116–131. Springer, 2012.
- [4] Riad Akrou, Marc Schoenauer, Michèle Sebag, and Jean-Christophe Souplet. Programming by feedback. In *International Conference on Machine Learning*, volume 32, pages 1503–1511. JMLR. org, 2014.
- [5] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.
- [6] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.
- [7] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- [8] Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Sinahal, and Anca D Dragan. Do you want your autonomous car to drive like you? In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 417–425. IEEE, 2017.
- [9] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96, 2005.
- [10] Christopher J Burges, Robert Ragno, and Quoc V Le. Learning to rank with nonsmooth cost functions. In *Advances in neural information processing systems*, pages 193–200, 2007.
- [11] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. Preference-based evolutionary direct policy search. In *ICRA Workshop on Autonomous Learning*, 2013.
- [12] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Advances in neural information processing systems*, 2011.
- [13] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.
- [14] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144. ACM, 2005.
- [15] Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Conference on Learning Theory (COLT)*, 2015.
- [16] Bianca Dumitrascu, Karen Feng, and Barbara Engelhardt. PG-TS: Improved Thompson sampling for logistic contextual bandits. In *Advances in Neural Information Processing Systems*, pages 4624–4633, 2018.

- [17] Brochu Eric, Nando D Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In *Advances in neural information processing systems*, pages 409–416, 2008.
- [18] Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. *Machine learning*, 89(1-2):123–156, 2012.
- [19] Subhashis Ghosal et al. Asymptotic normality of posterior distributions in high-dimensional linear models. *Bernoulli*, 5(2):315–331, 1999.
- [20] Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized Markov decision processes. In *Conference on Learning Theory*, pages 861–898, 2015.
- [21] Christian Gourieroux and Alain Monfort. Asymptotic properties of the maximum likelihood estimator in dichotomous logit models. *Journal of Econometrics*, 17(1):83–97, 1981.
- [22] R Herbrich, T Graepel, and K Obermayer. Support vector learning for ordinal regression. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99.(Conf. Publ. No. 470)*, volume 1, pages 97–102. IET, 1999.
- [23] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.
- [24] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.
- [25] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.
- [26] Andras Kupcsik, David Hsu, and Wee Sun Lee. Learning dynamic robot-to-human object handover from human feedback. In *Robotics research*, pages 161–176. Springer, 2018.
- [27] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.
- [28] Kevin P Murphy. *Machine learning: A probabilistic perspective*. The MIT Press, 2012.
- [29] Ian Osband, Daniel Russo, and Benjamin Van Roy. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.
- [30] Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2701–2710. JMLR. org, 2017.
- [31] Filip Radlinski and Thorsten Joachims. Query chains: Learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248. ACM, 2005.
- [32] Siddhartha Y Ramamohan, Arun Rajkumar, and Shivani Agarwal. Dueling bandits: Beyond condorcet winners to general tournament solutions. In *Advances in Neural Information Processing Systems*, pages 1253–1261, 2016.
- [33] Karthik Raman, Thorsten Joachims, Pannaga Shivaswamy, and Tobias Schnabel. Stable coactive learning via perturbation. In *International Conference on Machine Learning*, pages 837–845, 2013.
- [34] Carl Edward Rasmussen and Christopher K Williams. Gaussian processes for machine learning. *The MIT Press*, 2(3):4, 2006.
- [35] Diederik M Roijers, Luisa M Zintgraf, and Ann Nowé. Interactive Thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic Decision Theory*, pages 18–34. Springer, 2017.

- [36] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems (RSS)*, 2017.
- [37] Steven L Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.
- [38] Pannaga Shivaswamy and Thorsten Joachims. Online structured prediction via coactive learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 59–66. Omnipress, 2012.
- [39] Pannaga Shivaswamy and Thorsten Joachims. Coactive learning. *Journal of Artificial Intelligence Research*, 53:1–40, 2015.
- [40] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with Gaussian processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4781–4789. PMLR, 10–15 Jul 2018.
- [41] Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2017.
- [42] Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *IJCAI*, pages 5502–5510, 2018.
- [43] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [44] Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for Plackett-Luce: A dueling bandits approach. In *Advances in Neural Information Processing Systems*, pages 604–612, 2015.
- [45] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.
- [46] Aaron Wilson, Alan Fern, and Prasad Tadepalli. A Bayesian approach for policy learning from trajectory preference queries. In *Advances in neural information processing systems*, pages 1133–1141, 2012.
- [47] Christian Wirth. *Efficient Preference-based Reinforcement Learning*. PhD thesis, Technische Universität, 2017.
- [48] Christian Wirth, Riad Akrou, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18(1):4945–4990, 2017.
- [49] Christian Wirth and Johannes Fürnkranz. EPMC: Every visit preference Monte Carlo for reinforcement learning. In *Asian Conference on Machine Learning*, pages 483–497, 2013.
- [50] Christian Wirth and Johannes Fürnkranz. A policy iteration algorithm for learning from preference-based feedback. In *International Symposium on Intelligent Data Analysis*, pages 427–437. Springer, 2013.
- [51] Christian Wirth, Johannes Fürnkranz, and Gerhard Neumann. Model-free preference-based reinforcement learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [52] Huasen Wu and Xin Liu. Double Thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems*, pages 649–657, 2016.
- [53] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.
- [54] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278. ACM, 2007.

- [55] Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten De Rijke. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pages 307–315, 2015.
- [56] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten De Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *International Conference on Machine Learning (ICML)*, 2014.

## A1 Bayesian state/action credit assignment: two additional approaches

*Credit assignment* [48] is the problem of inferring which states or state/action pairs are responsible for observed user preferences. This paper derives theoretical guarantees for a Bayesian logistic regression credit assignment model. In this Appendix, we detail two additional credit assignment models—employing Bayesian linear regression and Gaussian process regression, respectively—for inferring a posterior over state/action utilities using trajectory preferences. (Note that these methods could similarly model utilities over states, rather than state/action pairs.)

In what follows,  $\tilde{s}$  denotes a state/action pair, with  $D = SA$  representing the number of possible values of  $\tilde{s}$ . For each trajectory  $\tau = \{\tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \dots, \tilde{s}_h\}$ , we observe the user’s preference, yielding a dataset  $\{\tau_i | i \in 1, \dots, N\}$  of  $N$  labeled trajectories. Define  $X \in \mathbb{R}^{N \times D}$  such that  $x_{ij} := [X]_{ij}$  is the number of times that trajectory  $i$  visits state/action  $\tilde{s}_j$ . Finally, we denote the label vector as  $\mathbf{y} \in \{0, 1\}^N$ , where the  $i^{\text{th}}$  element  $y_i$  is the preference label corresponding to  $\tau_i$ ; for instance, if  $\tau_1$  is preferred to  $\tau_2$ , then we would have  $y_1 = 1$  and  $y_2 = 0$ .

As before,  $\bar{r}(\tilde{s})$  represents the true state/action utilities, such that  $\bar{r}(\tau) = \sum_{i=1}^h \bar{r}(\tilde{s}_i)$ , with  $\bar{r}(\tau)$  being trajectory  $\tau$ ’s total (unobserved) utility. To apply regression methods to this data using preference labels, we approximate  $y_i \approx \bar{r}(\tau_i)$  to infer  $\bar{r}(\tilde{s})$ . In the following, we denote  $\hat{r}(\tilde{s})$  as our model of the true utilities  $\bar{r}(\tilde{s})$ . Also, define  $\hat{\mathbf{r}} \in \mathbb{R}^D$  as a vector in which the  $i^{\text{th}}$  element is  $\hat{r}(\tilde{s}_i)$ .

In the following, we discuss how to perform Bayesian inference on  $\bar{r}(\tilde{s})$  using  $\bar{r}(\tau_i)$ , which is approximated in practice via preferences. Note that the approximation  $y_i \approx \bar{r}(\tau_i)$  performs well empirically, though future work could perhaps apply Bayesian methods such as those in [14] to infer trajectories’ total utilities from the preference labels.

### A1.1 Bayesian linear regression

One can estimate state/action utilities from preferences via linear regression:  $\mathbf{y} = X\hat{\mathbf{r}} + \varepsilon$ , where  $\varepsilon$  is a vector of residuals and the other quantities are defined above. Bayesian linear regression infers a distribution over likely values of  $\hat{\mathbf{r}}$ . We define conjugate Gaussian prior and likelihood distributions over the state/action utilities and preference labels, respectively, to obtain a Gaussian posterior distribution over  $\hat{\mathbf{r}}$ . The prior, likelihood, and posterior take the following form, where  $\Lambda \in \mathbb{R}^{D \times D}$  and  $\sigma \in \mathbb{R}$  are prior parameters,  $\Lambda$  is positive definite, and we set  $\Lambda = \lambda \mathbb{I}$  for some  $\lambda > 0$ :

Prior:  $\hat{\mathbf{r}} \sim \mathcal{N}(0, \Lambda^{-1})$ ,  $\Lambda = \lambda \mathbb{I}$ ; Likelihood:

$$p(\mathbf{y} | X, \hat{\mathbf{r}}; \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - X\hat{\mathbf{r}}\|^2\right)$$

Posterior:  $\hat{\mathbf{r}} | X, \mathbf{y}, \sigma^2, \Lambda \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ , where: (12)

$$\boldsymbol{\mu} = (X^T X + \sigma^2 \Lambda)^{-1} X^T \mathbf{y} \quad (13)$$

$$\Sigma = \sigma^2 (X^T X + \sigma^2 \Lambda)^{-1} \quad (14)$$

### A1.2 Gaussian process regression

Gaussian process regression [34] extends Bayesian linear regression credit assignment to larger state and action spaces by generalizing across nearby states and actions. We will model the state/action utilities  $\hat{r}(\tilde{s})$  as a Gaussian process [34] over  $\tilde{s}$ , and then use the observed preferences to perform Gaussian process inference on  $\hat{\mathbf{r}}$ .

We model  $\hat{\mathbf{r}}$  as a Gaussian process:  $\hat{\mathbf{r}} \sim \mathcal{GP}(\boldsymbol{\mu}_r, K_r)$ , where  $\boldsymbol{\mu}_r$  is the prior mean vector and  $K_r$  is the prior covariance matrix, such that  $[K_r]_{ij}$  gives the prior covariance between  $\hat{r}(\tilde{s}_i)$  and  $\hat{r}(\tilde{s}_j)$ . We model  $\bar{r}(\tau_i)$ , the total utility for trajectory  $\tau_i$ , as a sum over the latent state/action utilities:

$$\bar{r}(\tau_i) = \sum_{j=1}^D x_{ij} \hat{r}(\tilde{s}_j) + \varepsilon_i, \quad (15)$$

with i.i.d. residuals  $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ . Because any linear combination of jointly Gaussian variables is Gaussian,  $\bar{r}(\tau_i)$  is a Gaussian process over the  $x_{ij}$ 's. Let  $\mathbf{R} \in \mathbb{R}^N$  be the vector with  $i^{\text{th}}$  element equal to  $R_i = \bar{r}(\tau_i)$ . Calculating the relevant expectations and covariances (see A1.3), one can show that  $\hat{\mathbf{r}}$  and  $\mathbf{R}$  are jointly normally distributed as follows:

$$\begin{bmatrix} \hat{\mathbf{r}} \\ \mathbf{R} \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \boldsymbol{\mu}_r \\ X\boldsymbol{\mu}_r \end{bmatrix}, \begin{bmatrix} K_r & K_r X^T \\ X K_r^T & X K_r X^T + \sigma_\varepsilon^2 \mathbb{I} \end{bmatrix} \right). \quad (16)$$

The standard approach for obtaining a conditional distribution from a joint Gaussian distribution [34] yields  $\hat{\mathbf{r}}|\mathbf{R} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ , where:

$$\boldsymbol{\mu} = \boldsymbol{\mu}_r + K_r X^T [X K_r X^T + \sigma_\varepsilon^2 \mathbb{I}]^{-1} (\mathbf{R} - X\boldsymbol{\mu}_r) \quad (17)$$

$$\Sigma = K_r - K_r X^T [X K_r X^T + \sigma_\varepsilon^2 \mathbb{I}]^{-1} X K_r^T. \quad (18)$$

In practice,  $\mathbf{R}$  is substituted with  $\mathbf{y}$  to perform the inference.

### A1.3 Using Gaussian processes to infer state/action utilities from preferences

This section derives the posterior inference equations (17) and (18), used in Gaussian process credit assignment. In this derivation, we act as though we observe the trajectories' total utilities  $\mathbf{R}$ , while remembering that in practice,  $\mathbf{R}$  is approximated via the user's preferences. Recall that  $\hat{\mathbf{r}} \in \mathbb{R}^D$  has  $i^{\text{th}}$  element  $\hat{r}(\tilde{s}_i)$ , which models the utility of state/action  $i$ , while  $\mathbf{R} \in \mathbb{R}^N$  has  $i^{\text{th}}$  element  $R_i = \bar{r}(\tau_i)$ . Let  $\mathbf{x}_i$  be the transpose of the  $i^{\text{th}}$  row of  $X$ .

Our goal is to infer the values of  $\hat{\mathbf{r}}$  given observations  $\mathbf{R}$  of the trajectories' total utilities. This can be accomplished via the following four steps:

1. Model the state/action utilities  $\hat{r}(\tilde{s})$  as a Gaussian process over  $\tilde{s}$ .
2. Model the trajectory utilities  $\mathbf{R}$  as a Gaussian process that can be defined as a sum of the state/action utilities  $\hat{r}(\tilde{s})$ .
3. Using the two Gaussian processes defined in 1) and 2), obtain the covariance matrix between the values of  $\{\hat{r}(\tilde{s})|\tilde{s} \in 1, \dots, D\}$  and  $\{R_i|i \in 1, \dots, N\}$ .
4. Write the joint Gaussian distribution between the values of  $\{\hat{r}(\tilde{s})|\tilde{s} \in 1, \dots, D\}$  and  $\{R_i|i \in 1, \dots, N\}$ , and obtain the posterior distribution of  $\hat{\mathbf{r}}$  at all state/actions given  $\mathbf{R}$ .

The four subsequent subsections correspond to these four steps, respectively.

#### A1.3.1 The state/action utility Gaussian process

We model the state/action utilities as a Gaussian process over  $\tilde{s}$ , with mean  $\mathbb{E}[\hat{r}(\tilde{s})] = \mu_r(\tilde{s})$  and covariance kernel  $\text{Cov}(\hat{r}(\tilde{s}), \hat{r}(\tilde{s}')) = k_r(\tilde{s}, \tilde{s}')$ , for all state/action pairs  $\tilde{s}, \tilde{s}'$ . Thus,

$$\hat{r}(\tilde{s}) \sim \mathcal{GP}(\mu_r(\tilde{s}), k_r(\tilde{s}, \tilde{s}')). \quad (19)$$

Define  $\boldsymbol{\mu}_r \in \mathbb{R}^D$  such that the  $i^{\text{th}}$  element is  $[\boldsymbol{\mu}_r]_i = \mu_r(\tilde{s}_i)$ , the prior mean for the utility of state/action  $\tilde{s}_i$ . Let  $K_r \in \mathbb{R}^{D \times D}$  be the covariance matrix over state/action utilities, such that  $[K_r]_{ij} = k_r(\tilde{s}_i, \tilde{s}_j)$ . Therefore,  $\hat{\mathbf{r}}$ , the vector for which  $[\hat{\mathbf{r}}]_i = \hat{r}(\tilde{s}_i)$ , is also a Gaussian process:

$$\hat{\mathbf{r}} \sim \mathcal{GP}(\boldsymbol{\mu}_r, K_r). \quad (20)$$

#### A1.3.2 The trajectory utility Gaussian process

By assumption, the trajectory utilities  $\mathbf{R} \in \mathbb{R}^N$  are sums of the latent state/action utilities via the following relationship between  $\mathbf{R}$  and  $\hat{\mathbf{r}}$ :

$$R(\mathbf{x}_i) := R_i = \sum_{j=1}^D x_{ij} \hat{r}(\tilde{s}_j) + \varepsilon_i, \quad (21)$$

where  $\varepsilon_i$  are i.i.d. Gaussian noise variables with distribution  $\mathcal{N}(0, \sigma_\varepsilon^2)$ .

Note that  $R(\mathbf{x}_i)$  is a Gaussian process over  $\mathbf{x}_i \in \mathbb{R}^D$  because  $\{\hat{r}(\tilde{s}), \forall \tilde{s}\}$  are jointly normally distributed by definition of a Gaussian process, and any linear combination of jointly Gaussian variables has a univariate normal distribution.

Next, we calculate the expectation and covariance of  $\mathbf{R}$  over the observations. The expectation of its  $i^{\text{th}}$  element  $R_i = R(\mathbf{x}_i)$  can be expressed as follows:

$$\mathbb{E}[R_i] = \mathbb{E}\left[\sum_{j=1}^D x_{ij} \hat{r}(\tilde{s}_j) + \varepsilon_i\right] = \sum_{j=1}^D x_{ij} \mathbb{E}[\hat{r}(\tilde{s}_j)] = \sum_{j=1}^D x_{ij} \mu_r(\tilde{s}_j). \quad (22)$$

The expectation over  $\mathbf{R}(X)$  can thus be written as:

$$\mathbb{E}[\mathbf{R}(X)] = X \boldsymbol{\mu}_r. \quad (23)$$

Next, we model the covariance matrix of  $\mathbf{R}(X)$ . The  $ij^{\text{th}}$  element of this matrix is the covariance of  $R(\mathbf{x}_i)$  and  $R(\mathbf{x}_j)$ :

$$\text{Cov}(R(\mathbf{x}_i), R(\mathbf{x}_j)) = \mathbb{E}[R(\mathbf{x}_i)R(\mathbf{x}_j)] - \mathbb{E}[R(\mathbf{x}_i)]\mathbb{E}[R(\mathbf{x}_j)] \quad (24)$$

$$= \mathbb{E}[R(\mathbf{x}_i)R(\mathbf{x}_j)] - \left(\sum_{k=1}^D x_{ik} \mu_r(\tilde{s}_k)\right) \left(\sum_{m=1}^D x_{jm} \mu_r(\tilde{s}_m)\right) \quad (25)$$

$$= \mathbb{E}\left[\left(\sum_{k=1}^D x_{ik} \hat{r}(\tilde{s}_k) + \varepsilon_i\right) \left(\sum_{m=1}^D x_{jm} \hat{r}(\tilde{s}_m) + \varepsilon_j\right)\right] \quad (26)$$

$$- \left(\sum_{k=1}^D x_{ik} \mu_r(\tilde{s}_k)\right) \left(\sum_{m=1}^D x_{jm} \mu_r(\tilde{s}_m)\right) \quad (27)$$

$$= \sum_{k=1}^D \sum_{m=1}^D x_{ik} x_{jm} \mathbb{E}[\hat{r}(\tilde{s}_k) \hat{r}(\tilde{s}_m)] + \mathbb{E}[\varepsilon_i \varepsilon_j] - \sum_{k=1}^D \sum_{m=1}^D x_{ik} x_{jm} \mu_r(\tilde{s}_k) \mu_r(\tilde{s}_m) \quad (28)$$

$$= \sum_{k=1}^D \sum_{m=1}^D x_{ik} x_{jm} [\text{Cov}(\hat{r}(\tilde{s}_k), \hat{r}(\tilde{s}_m)) + \mu_r(\tilde{s}_k) \mu_r(\tilde{s}_m)] \quad (29)$$

$$- \sum_{k=1}^D \sum_{m=1}^D x_{ik} x_{jm} \mu_r(\tilde{s}_k) \mu_r(\tilde{s}_m) + \sigma_\varepsilon^2 \mathbb{I}_{[i=j]} \quad (30)$$

$$= \sum_{k=1}^D \sum_{m=1}^D x_{ik} x_{jm} \text{Cov}(\hat{r}(\tilde{s}_k), \hat{r}(\tilde{s}_m)) + \sigma_\varepsilon^2 \mathbb{I}_{[i=j]} \quad (31)$$

$$= \sum_{k=1}^D \sum_{m=1}^D x_{ik} x_{jm} k_r(\tilde{s}_k, \tilde{s}_m) + \sigma_\varepsilon^2 \mathbb{I}_{[i=j]} = \mathbf{x}_i^T K_r \mathbf{x}_j + \sigma_\varepsilon^2 \mathbb{I}_{[i=j]}. \quad (32)$$

We can then write the covariance matrix of  $\mathbf{R}$  as  $K_R(X)$ , where:

$$[K_R(X)]_{ij} = \text{Cov}(R(\mathbf{x}_i), R(\mathbf{x}_j)) = \mathbf{x}_i^T K_r \mathbf{x}_j + \sigma_\varepsilon^2 \mathbb{I}_{[i=j]}. \quad (33)$$

From here, it can be seen that  $K_R(X) = X K_r X^T + \sigma_\varepsilon^2 \mathbb{I}$ :

$$X K_r X^T = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} K_r \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} \begin{bmatrix} K_r \mathbf{x}_1 & K_r \mathbf{x}_2 & \dots & K_r \mathbf{x}_N \end{bmatrix} \quad (34)$$

$$= \begin{bmatrix} \mathbf{x}_1^T K_r \mathbf{x}_1 & \dots & \mathbf{x}_1^T K_r \mathbf{x}_N \\ \vdots & \ddots & \vdots \\ \mathbf{x}_N^T K_r \mathbf{x}_1 & \dots & \mathbf{x}_N^T K_r \mathbf{x}_N \end{bmatrix} = K_R(X) - \sigma_\varepsilon^2 \mathbb{I}. \quad (35)$$

### A1.3.3 Covariance between state/action and trajectory utilities

In this subsection, we consider the covariance between  $\hat{\mathbf{r}}$  and  $\mathbf{R}$ , denoted  $K_{\hat{\mathbf{r}}, \mathbf{R}}$ :

$$[K_{\hat{\mathbf{r}}, \mathbf{R}}]_{ij} = \text{Cov}([\hat{\mathbf{r}}]_i, [\mathbf{R}]_j) = \text{Cov}(\hat{r}(\tilde{s}_i), R(\mathbf{x}_j)). \quad (36)$$

This covariance matrix can be expressed in terms of  $X$ ,  $K_r$ , and  $\boldsymbol{\mu}_r$ :

$$[K_{\hat{\mathbf{r}}, \mathbf{R}}]_{ij} = \text{Cov}(\hat{r}(\tilde{s}_i), R(\mathbf{x}_j)) = \text{Cov}\left(\hat{r}(\tilde{s}_i), \sum_{k=1}^D x_{jk} \hat{r}(\tilde{s}_k) + \varepsilon_j\right) \quad (37)$$

$$= \mathbb{E}\left[\hat{r}(\tilde{s}_i) \sum_{k=1}^D x_{jk} \hat{r}(\tilde{s}_k) + \varepsilon_j \hat{r}(\tilde{s}_i)\right] - \mathbb{E}[\hat{r}(\tilde{s}_i)] \mathbb{E}\left[\sum_{k=1}^D x_{jk} \hat{r}(\tilde{s}_k) + \varepsilon_j\right] \quad (38)$$

$$= \sum_{k=1}^D x_{jk} \mathbb{E}[\hat{r}(\tilde{s}_i) \hat{r}(\tilde{s}_k)] - [\boldsymbol{\mu}_r(\tilde{s}_i)] [\mathbf{x}_j^T \boldsymbol{\mu}_r] \quad (39)$$

$$= \sum_{k=1}^D x_{jk} \{\text{Cov}(\hat{r}(\tilde{s}_i), \hat{r}(\tilde{s}_k)) + \mathbb{E}[\hat{r}(\tilde{s}_i)] \mathbb{E}[\hat{r}(\tilde{s}_k)]\} - \mu_r(\tilde{s}_i) \mathbf{x}_j^T \boldsymbol{\mu}_r \quad (40)$$

$$= \sum_{k=1}^D x_{jk} [k_r(\tilde{s}_i, \tilde{s}_k) + \mu_r(\tilde{s}_i) \mu_r(\tilde{s}_k)] - \mu_r(\tilde{s}_i) \mathbf{x}_j^T \boldsymbol{\mu}_r \quad (41)$$

$$= \sum_{k=1}^D x_{jk} k_r(\tilde{s}_i, \tilde{s}_k) + \mu_r(\tilde{s}_i) \mathbf{x}_j^T \boldsymbol{\mu}_r - \mu_r(\tilde{s}_i) \mathbf{x}_j^T \boldsymbol{\mu}_r = \sum_{k=1}^D x_{jk} k_r(\tilde{s}_i, \tilde{s}_k) = \mathbf{x}_j^T [K_r]_{i,:}, \quad (42)$$

where  $[K_r]_{i,:}$  is the column vector obtained by transposing the  $i^{\text{th}}$  row of  $K_r$ . From here, one can see that  $K_{\hat{\mathbf{r}}, \mathbf{R}} = K_r X^T$ :

$$K_r X^T = \begin{bmatrix} [K_r]_{1,:}^T \\ [K_r]_{2,:}^T \\ \vdots \\ [K_r]_{D,:}^T \end{bmatrix} * [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_N] = K_{\hat{\mathbf{r}}, \mathbf{R}}. \quad (43)$$

### A1.3.4 Posterior inference over state/action utilities

The results of the previous three subsections can be combined to obtain the following joint probability density between  $\hat{\mathbf{r}}$  and  $\mathbf{R}$ :

$$\begin{bmatrix} \hat{\mathbf{r}} \\ \mathbf{R} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_r \\ X \boldsymbol{\mu}_r \end{bmatrix}, \begin{bmatrix} K_r & K_r X^T \\ X K_r^T & X K_r X^T + \sigma_\varepsilon^2 \mathbb{I} \end{bmatrix}\right). \quad (44)$$

This relationship expresses all components of the joint Gaussian density in terms of  $X$ ,  $K_r$ , and  $\boldsymbol{\mu}_r$ , or in other words, in terms of the state/action visit counts in the observed trajectories (captured by  $X$ ) and the Gaussian process prior on  $\hat{\mathbf{r}}$ .

Using the standard approach for obtaining a conditional distribution from a jointly Gaussian distribution (see Appendix A.2 in [34]), we arrive at:

$$\hat{\mathbf{r}} | \mathbf{R} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma), \text{ where:} \quad (45)$$

$$\boldsymbol{\mu} = \boldsymbol{\mu}_r + K_r X^T [X K_r X^T + \sigma_\varepsilon^2 \mathbb{I}]^{-1} (\mathbf{R} - X \boldsymbol{\mu}_r) \quad (46)$$

$$\Sigma = K_r - K_r X^T [X K_r X^T + \sigma_\varepsilon^2 \mathbb{I}]^{-1} X K_r^T. \quad (47)$$

Thus, we have expressed the conditional posterior density of  $\hat{\mathbf{r}}$  in terms of  $X$ ,  $K_r$ ,  $\boldsymbol{\mu}_r$ , and the observations  $\mathbf{R} \approx \mathbf{y}$ .



## A2 Proofs

This section proves the theoretical results stated in Section 5 of the paper.

We begin by restating a result from Gourieroux and Monfort [21] establishing conditions in which logistic regression is asymptotically consistent, after defining two necessary conditions.

**Definition 1** (Derivative of sigmoid).  $f : \mathbb{R} \rightarrow \mathbb{R}$ , where  $f = \frac{e^{-x}}{(1+e^{-x})^2}$ . Note that  $f(x) = f(-x)$ . This is the derivative of the sigmoid function,  $\sigma(x) = \frac{1}{1+e^{-x}}$ .

**Definition 2.** Let  $\bar{\mathbf{r}} \in \mathbb{R}^D$  be the vector of true state/action utilities (assumed to exist) and  $\bar{\mathbf{r}}' \in \mathbb{R}^{D-1}$  be its transformation via (4). Define  $\tilde{\mathbf{r}}'_k \in \mathbb{R}^{D-1}$  as the state/action rewards sampled from the Bayesian logistic regression model posterior in episode  $k$ ,  $\hat{\mathbf{r}}'_k \in \mathbb{R}^{D-1}$  as the model's maximum a posteriori (MAP) estimate at episode  $k$ , and  $\hat{\mathbf{r}}'_{ML,k} \in \mathbb{R}^{D-1}$  as its maximum likelihood estimate at  $k$ . Lastly,  $\tilde{\mathbf{r}}_k \in \mathbb{R}^D$ ,  $\hat{\mathbf{r}}_k \in \mathbb{R}^D$ , and  $\hat{\mathbf{r}}_{ML,k} \in \mathbb{R}^D$  are their respective equivalents given by (5).

**Condition 1.**  $\exists m_0 < \infty$  such that  $|z_{ij}| \leq m_0$  for all  $i \in \{1, \dots, N\}, j \in \{1, \dots, D-1\}$ .

Condition 1 is always satisfied because  $z_{ij} = \mathbf{x}_i^T \mathbf{v}_j$  by (4), where  $\mathbf{v}_j$  is a unit vector. Additionally,  $\mathbf{x}_i = \mathbf{x}_{1i} - \mathbf{x}_{i2}$  is difference of two vectors that both count how many times each state/action pair is visited in an episode, and thus both have only positive elements that sum to the episode horizon,  $h$ . So,  $|z_{ij}| \leq \|\mathbf{x}_i\|_2 \|\mathbf{v}_j\|_2 = \|\mathbf{x}_i\|_2 \leq \|\mathbf{x}_{1i} - \mathbf{x}_{i2}\|_1 = \|\mathbf{x}_{1i}\|_1 + \|\mathbf{x}_{i2}\|_1 = 2h$ , where the inequalities are respectively the Cauchy-Schwarz inequality,  $\|\mathbf{x}\|_p < \|\mathbf{x}\|_q$  for  $p > q > 0$ , and the triangle inequality. So, Condition 1 holds for  $m_0 = 2h$ .

**Condition 2.** Let  $\lambda_1^{(k)}$  and  $\lambda_{D-1}^{(k)}$  be the largest and smallest eigenvalues, respectively, of  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$ . Then,  $\exists m_1 < \infty$  such that  $\frac{\lambda_1^{(k)}}{\lambda_{D-1}^{(k)}} < m_1$ , for all  $k$ .

Intuitively, Condition 2 requires that an arbitrarily-high fraction of observations *cannot* lie in a proper subspace of their possible span. This condition is necessary, as otherwise data outside this subspace would become increasingly ignored as more data points are obtained. Condition 2 states that the inverse Hessian of the negative log reward posterior—evaluated at the true rewards  $\bar{\mathbf{r}}'$ —has an upper-bounded ratio between its largest and smallest eigenvalues. This can be explicitly enforced by bounding this maximum-to-minimum eigenvalue ratio for the matrix  $\sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k) \mathbf{z}_i \mathbf{z}_i^T$ , and only updating the reward posterior when the eigenvalue ratio is below-threshold. As shown in Lemma 1 below, satisfying this restriction bounds the eigenvalue ratio for  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$ , as desired.

Condition 2 may also be guaranteed in certain situations, e.g., if randomness in the dynamics is sufficient to ensure a full-row-rank observation matrix  $Z$ . In this case, the eigenvalue ratio will be bounded regardless of the specific policies executed. If properties of the environment are known to guarantee Condition 2, then it need not be explicitly enforced. Meanwhile, weakening the requirements of Condition 2—for instance, by strengthening the Bayesian logistic regression convergence analysis or considering other credit assignment models—would be an interesting avenue for future work.

**Lemma 1** (Enforcing Condition 2). *The eigenvalue ratio  $\frac{\lambda_{\max}(\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T)}{\lambda_{\min}(\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T)}$  has a constant upper bound that holds for all  $k$  if and only if the eigenvalue ratio  $\frac{\lambda_{\max}(\sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k) \mathbf{z}_i \mathbf{z}_i^T)}{\lambda_{\min}(\sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k) \mathbf{z}_i \mathbf{z}_i^T)}$  has a constant upper bound that holds for all  $k$ . Therefore, one can ensure that the former condition holds by enforcing the latter.*

*Proof.* We apply the result in Lemma 1.1 below. To verify that the conditions for this result hold, we must show that both  $f(\mathbf{z}_i^T \bar{\mathbf{r}}')$  and  $f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)$  have upper and lower bounds for all  $i, k$ , where the lower bound exceeds zero. The upper bound exists because  $f(x) \in (0, 1] \forall x$ .

It remains to show that both  $f(\mathbf{z}_i^T \bar{\mathbf{r}}')$  and  $f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)$  are lower-bounded above zero. Because  $f(x)$  monotonically decreases as  $|x|$  increases, this is true as long as  $|\mathbf{z}_i^T \bar{\mathbf{r}}'|$  and  $|\mathbf{z}_i^T \hat{\mathbf{r}}'_k|$  are upper-bounded. In the former case,  $|\mathbf{z}_i^T \bar{\mathbf{r}}'| < \|\mathbf{z}_i\|_2 \|\bar{\mathbf{r}}'\|_2$ . The quantity  $\|\mathbf{z}_i\|_2$  is upper-bounded by Condition 1. The rewards  $\bar{\mathbf{r}}'$  produce the same policy when scaled by any positive quantity, and so their magnitude can be viewed as fixed. The same logic holds in the latter case.

□

**Lemma 1.1** (Bounded eigenvalue ratios). *Let  $A_k, B_k \in \mathbb{R}^{n \times n}$  be two matrices of the form  $A_k = \sum_{i=1}^k \alpha_i \mathbf{v}_i \mathbf{v}_i^T$ ,  $B_k = \sum_{i=1}^k \beta_i \mathbf{v}_i \mathbf{v}_i^T$ , where  $\alpha_i \in [\alpha_{\min}, \alpha_{\max}]$ ,  $\beta_i \in [\beta_{\min}, \beta_{\max}]$ ,  $\alpha_{\min} > 0$ ,  $\beta_{\min} > 0$ , and  $\mathbf{v}_i \in \mathbb{R}^n$  for  $i \in \{1, \dots, k\}$ . Let  $\lambda_{\max}(M)$  and  $\lambda_{\min}(M)$  respectively be the largest eigenvalues of a matrix  $M$ . Then,  $\frac{\lambda_{\max}(A_k)}{\lambda_{\min}(A_k)}$  is upper-bounded for all  $T$  if and only if  $\frac{\lambda_{\max}(B_k)}{\lambda_{\min}(B_k)}$  is upper-bounded for all  $T$ .*

*Proof.* Without loss of generality, we assume that  $\frac{\lambda_{\max}(A_k)}{\lambda_{\min}(A_k)} < m_1 < \infty$  for some  $m_1$  and for all  $k$ , and will show that  $\frac{\lambda_{\max}(B_k)}{\lambda_{\min}(B_k)}$  has an upper bound for all  $k$ . Note that  $a\mathbf{u}\mathbf{u}^T \succeq 0$  (i.e., is positive semidefinite) for any  $a > 0$  and vector  $\mathbf{u}$ , and that sums of positive semidefinite matrices are also positive semidefinite. In addition, we will use the following facts about arbitrary positive semidefinite matrices  $A, B \succeq 0$  (which can be shown from the Courant-Fischer-Weyl min-max principle):

$$\lambda_{\max}(A) \leq \lambda_{\max}(A + B) \quad (48)$$

$$\lambda_{\min}(A) \leq \lambda_{\min}(A) + \lambda_{\min}(B) \leq \lambda_{\min}(A + B) \quad (49)$$

The desired result is an outcome of the following four relations:

$$\lambda_{\max}(A_k) = \lambda_{\max}\left(\sum_{i=1}^k \alpha_i \mathbf{v}_i \mathbf{v}_i^T\right) = \lambda_{\max}\left(\sum_{i=1}^k (\alpha_i - \alpha_{\min}) \mathbf{v}_i \mathbf{v}_i^T + \sum_{i=1}^k \alpha_{\min} \mathbf{v}_i \mathbf{v}_i^T\right) \quad (50)$$

$$\geq \lambda_{\max}\left(\sum_{i=1}^k \alpha_{\min} \mathbf{v}_i \mathbf{v}_i^T\right) = \alpha_{\min} \lambda_{\max}\left(\sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T\right), \text{ by (48)} \quad (51)$$

$$\lambda_{\min}(A_k) = \lambda_{\min}\left(\sum_{i=1}^k \alpha_i \mathbf{v}_i \mathbf{v}_i^T\right) \leq \lambda_{\min}\left(\sum_{i=1}^k \alpha_i \mathbf{v}_i \mathbf{v}_i^T\right) + \lambda_{\min}\left(\sum_{i=1}^k (\alpha_{\max} - \alpha_i) \mathbf{v}_i \mathbf{v}_i^T\right) \quad (52)$$

$$\leq \lambda_{\min}\left(\sum_{i=1}^k \alpha_{\max} \mathbf{v}_i \mathbf{v}_i^T\right) = \alpha_{\max} \lambda_{\min}\left(\sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T\right), \text{ by (49)} \quad (53)$$

$$\lambda_{\max}(B_k) = \lambda_{\max}\left(\sum_{i=1}^k \beta_i \mathbf{v}_i \mathbf{v}_i^T\right) \leq \lambda_{\max}\left(\sum_{i=1}^k \beta_i \mathbf{v}_i \mathbf{v}_i^T + \sum_{i=1}^k (\beta_{\max} - \beta_i) \mathbf{v}_i \mathbf{v}_i^T\right), \text{ by (48)} \quad (54)$$

$$= \lambda_{\max}\left(\sum_{i=1}^k \beta_{\max} \mathbf{v}_i \mathbf{v}_i^T\right) = \beta_{\max} \lambda_{\max}\left(\sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T\right) \quad (55)$$

$$\lambda_{\min}(B_k) = \lambda_{\min}\left(\sum_{i=1}^k \beta_i \mathbf{v}_i \mathbf{v}_i^T\right) = \lambda_{\min}\left(\sum_{i=1}^k (\beta_i - \beta_{\min}) \mathbf{v}_i \mathbf{v}_i^T + \sum_{i=1}^k \beta_{\min} \mathbf{v}_i \mathbf{v}_i^T\right) \quad (56)$$

$$\geq \lambda_{\min}\left(\sum_{i=1}^k \beta_{\min} \mathbf{v}_i \mathbf{v}_i^T\right) = \beta_{\min} \lambda_{\min}\left(\sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T\right), \text{ by (49)} \quad (57)$$

Now, we can upper bound the eigenvalue ratio for  $B_k$ :

$$\frac{\lambda_{\max}(B_k)}{\lambda_{\min}(B_k)} \leq \frac{\beta_{\max} \lambda_{\max}\left(\sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T\right)}{\beta_{\min} \lambda_{\min}\left(\sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^T\right)}, \text{ by (55) and (57)} \quad (58)$$

$$\leq \frac{\beta_{\max}}{\beta_{\min}} \frac{\lambda_{\max}(A_k)}{\alpha_{\min}} \left[ \frac{\lambda_{\min}(A_k)}{\alpha_{\max}} \right]^{-1}, \text{ by (51) and (53)} \quad (59)$$

$$= \frac{\beta_{\max} \alpha_{\max}}{\beta_{\min} \alpha_{\min}} \frac{\lambda_{\max}(A_k)}{\lambda_{\min}(A_k)} \leq \frac{\beta_{\max} \alpha_{\max}}{\beta_{\min} \alpha_{\min}} * m_1. \quad (60)$$

□

We will apply the following result from Gouriou and Monfort [21] concerning asymptotic consistency of the maximum likelihood estimator:

**Proposition 1** (Asymptotic consistency of logistic regression [21]). *If Conditions 1 and 2 are satisfied, then the maximum likelihood estimator  $\hat{\mathbf{r}}'_{ML,k}$  of  $\bar{\mathbf{r}}'$  exists almost surely as  $k \rightarrow \infty$ , and  $\hat{\mathbf{r}}'_{ML,k}$  converges almost surely to the true value  $\bar{\mathbf{r}}'$  if and only if  $\lim_{k \rightarrow \infty} \lambda_{D-1}^{(k)} = \infty$ .*

*Proof.* This result is a restatement of Proposition 3 in Gouriou and Monfort [21].  $\square$

**Remark:** The proof of Proposition 1 in [21] can be adapted such that the same result holds when the maximum likelihood estimator  $\hat{\mathbf{r}}'_{ML,k}$  is replaced with the MAP estimator  $\hat{\mathbf{r}}'_k$ ; thus, it applies to our setting. We show that Proposition 1's final condition for convergence of the MAP estimator holds:

**Lemma 2.** *Under known transition dynamics, all eigenvalues of the matrix  $\sum_{i=1}^k f(z_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$  approach infinity as  $k \rightarrow \infty$ .*

*Proof.* Let  $\alpha_i = f(z_i^T \bar{\mathbf{r}}')$  and  $\hat{\alpha}_i^{(k)} = f(y_i z_i^T \hat{\mathbf{r}}'_k)$ . The values  $\alpha_i$  are both upper-bounded and non-decaying:  $f(x) \in (0, 1)$  for all  $x$ , and  $\mathbf{z}_i^T \bar{\mathbf{r}}'$  is bounded in magnitude since we assume the true rewards  $\bar{\mathbf{r}}'$  are bounded. We can write:

$$\sum_{i=1}^N f(z_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T = \sum_{i=1}^N \alpha_i \mathbf{z}_i \mathbf{z}_i^T = \begin{bmatrix} \sqrt{\alpha_1} \mathbf{z}_1 & \sqrt{\alpha_2} \mathbf{z}_2 & \dots & \sqrt{\alpha_N} \mathbf{z}_N \end{bmatrix} \begin{bmatrix} \sqrt{\alpha_1} \mathbf{z}_1^T \\ \sqrt{\alpha_2} \mathbf{z}_2^T \\ \vdots \\ \sqrt{\alpha_N} \mathbf{z}_N^T \end{bmatrix}. \quad (61)$$

Define  $M \in \mathbb{R}^{N \times (D-1)}$ ,  $M_1 \in \mathbb{R}^{m \times (D-1)}$ , and  $M_2 \in \mathbb{R}^{(N-m) \times (D-1)}$  as:

$$M = \begin{bmatrix} \sqrt{\alpha_1} \mathbf{z}_1^T \\ \sqrt{\alpha_2} \mathbf{z}_2^T \\ \vdots \\ \sqrt{\alpha_N} \mathbf{z}_N^T \end{bmatrix}, \quad M_1 = \begin{bmatrix} \sqrt{\alpha_1} \mathbf{z}_1^T \\ \sqrt{\alpha_2} \mathbf{z}_2^T \\ \vdots \\ \sqrt{\alpha_m} \mathbf{z}_m^T \end{bmatrix}, \quad M_2 = \begin{bmatrix} \sqrt{\alpha_{m+1}} \mathbf{z}_{m+1}^T \\ \vdots \\ \sqrt{\alpha_N} \mathbf{z}_N^T \end{bmatrix}. \quad (62)$$

Then,

$$M = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}, \quad \text{and} \quad M^T M = \sum_{i=1}^N \alpha_i \mathbf{z}_i \mathbf{z}_i^T = M_1^T M_1 + M_2^T M_2. \quad (63)$$

Also, define  $\hat{M} \in \mathbb{R}^{N \times (D-1)}$ ,  $\hat{M}_1 \in \mathbb{R}^{m \times (D-1)}$ , and  $\hat{M}_2 \in \mathbb{R}^{(N-m) \times (D-1)}$  analogously, but replacing  $\alpha_i$  with  $\hat{\alpha}_i^{(N)}$ . Note that  $M^T M$  and  $\hat{M}^T \hat{M}$  have the same rank, since  $M$  and  $\hat{M}$  have the same row-rank; similarly  $M^T M_i$  and  $\hat{M}_i^T \hat{M}_i$  have the same rank for  $i \in \{1, 2\}$ .

Assume that there exists  $m \in \mathbb{N}$  such that  $\hat{M}_2^T \hat{M}_2$  has rank less than  $(D-1)$ ; let  $r \geq 1$  be the number of zero eigenvalues of  $\hat{M}_2^T \hat{M}_2$ . We will show that as  $N$  increases, the probability of drawing a sample  $\mathbf{z}_i$  that is linearly independent of the rows already in  $\hat{M}_2$  (that is, a sample that increases the rank of  $\hat{M}_2^T \hat{M}_2$ ) does not decay to zero. This means that there will be infinitely-many non-overlapping sets of indices  $\{j, \dots, k\}$  such that  $\sum_{i=j}^k \alpha_i \mathbf{z}_i \mathbf{z}_i^T$  has rank  $(D-1)$ . From here, we can show that  $(D-1)$  of the eigenvalues of  $M^T M$  approach infinity (recall that the  $\alpha_i$  values are lower-bounded).

Under the Laplace approximation, the posterior covariance after  $N$  episodes takes the form  $\Sigma_N = \left( \sum_{i=1}^N f(y_i \mathbf{z}_i^T \hat{\mathbf{r}}_N) \mathbf{z}_i \mathbf{z}_i^T + V_0^{-1} \right)^{-1} = \left( \hat{M}^T \hat{M} + V_0^{-1} \right)^{-1}$ . Under our assumption that  $\hat{M}_2^T \hat{M}_2$  has  $r \geq 1$  eigenvalues equal to zero, the row-rank of  $\hat{M}_2$  is  $(D-1-r)$ .

We show that  $\Sigma_N$  has  $r$  eigenvalues that are bounded away from zero. We write the eigenvalues of any square matrix  $M_0 \in \mathbb{R}^{n \times n}$  as  $\lambda_1(M_0) \geq \lambda_2(M_0) \geq \dots \geq \lambda_n(M_0)$ . Then, we apply the

following fact (which can be shown from the Courant-Fischer-Weyl min-max principle): for positive semidefinite matrices  $A, B \in \mathbb{R}^{n \times n}$ ,

$$\lambda_n(A) + \lambda_k(B) \leq \lambda_k(A + B) \leq \lambda_1(A) + \lambda_k(B). \quad (64)$$

Clearly,  $\sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T$  and  $V_0^{-1}$  are both positive semidefinite ( $V_0$  is positive definite by its definition), and so:

$$0 \leq \lambda_k \left( \sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T + V_0^{-1} \right) \leq \lambda_1(V_0^{-1}) + \lambda_k \left( \sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T \right), \quad (65)$$

where the first inequality holds because the sum of two positive semidefinite matrices is also positive semidefinite, and the second inequality is an application of the inequality in (64). The first term in (65) is a constant, while the second is zero for  $k > D - 1 - r$ . Thus,  $\Sigma_N^{-1} = \sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T + V_0^{-1}$  has  $r$  eigenvalues that are upper-bounded, where the upper bound depends only on  $V_0$ . Therefore,  $\Sigma$  has  $r$  eigenvalues that are lower-bounded by  $[\lambda_1(V_0^{-1})]^{-1} = \left[ \frac{1}{\lambda_{D-1}(V_0)} \right]^{-1} = \lambda_{D-1}(V_0) > 0$ ; the other  $D - 1 - r$  eigenvalues of  $\Sigma$  all decay to zero.

To complete the proof, we argue that the following statements hold; the first two of these statements are proven in the two subsequent sub-lemmas; the third statement completes the proof.

1. Note that  $M^T M$  and  $\hat{M}^T \hat{M}$  have the same rank, due to  $M$  and  $\hat{M}$  having the same row-rank. Assume that for some  $m$ ,  $\hat{M}_2^T \hat{M}_2$  has  $r \geq 1$  zero eigenvalues. The reward vector  $\tilde{\mathbf{r}}'$  sampled from the logistic regression posterior can be expressed in terms of the eigenbasis of  $\hat{M}_2^T \hat{M}_2$ ,  $\{\boldsymbol{\nu}_i | i = 1, \dots, D - 1\}$ :  $\tilde{\mathbf{r}}' = \sum_{i=1}^{D-1} \beta_i \boldsymbol{\nu}_i$  for some coefficients  $\beta_i$ . Then, as  $N \rightarrow \infty$ , for  $j$  such that  $\mathbf{v}_j$  is in the null-space of  $\hat{M}_2^T \hat{M}_2$ , the probability of sampling  $\tilde{\mathbf{r}}'_N$  such that  $\frac{|\beta_j|}{\sum_{i=1}^{D-1} |\beta_i|} \geq b$  for any  $b \in (0, 1)$  does not decay to zero.
2. There exists  $b \in (0, 1)$  such that if  $\frac{\beta_j}{\sum_{i=1}^{D-1} \beta_i} \geq b$ , then with probability above zero, the policy will sample a trajectory such that the next  $\mathbf{z}_i$  sample has a nonzero component in  $\boldsymbol{\nu}_j$ .
3. If a trajectory as described in 2) is sampled, a zero-eigenvalue of  $M_2^T M_2$  increases by a lower-bounded amount. This comes from the fact that there are only finitely-many possible  $\mathbf{z}_i$  vectors, so if  $\mathbf{z}_i$  has nonzero inner product with an eigenvector, this inner product cannot be arbitrarily close to zero. This contradicts that  $\text{Rank}(M_2^T M_2)$  remains below  $D - 1$  as  $N \rightarrow \infty$ .

□

**Lemma 2.1** (Proof of Statement 1 in Lemma 2).

*Proof.* In episode  $N$ , the sampled reward vector  $\tilde{\mathbf{r}}'_N$  is drawn from the logistic regression posterior,  $\mathcal{N}(\hat{\mathbf{r}}', \Sigma_N)$ , with covariance matrix  $\Sigma_N = \left( \sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T + V_0^{-1} \right)^{-1}$ . Consider a sample  $\tilde{\mathbf{r}}'_N$  with a multivariate normal distribution in  $\mathbb{R}^{D-1}$ :

$$\mathcal{N}(\tilde{\mathbf{r}}'_N | \hat{\mathbf{r}}'_N, \Sigma_N) \propto \exp \left( -\frac{1}{2} (\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N)^T \Sigma_N^{-1} (\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N) \right). \quad (66)$$

The intuition for this proof is as follows. There is a decreasing probability of sampling a vector  $\tilde{\mathbf{r}}'_N$  such that  $(\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N)$  has components in eigenvectors of  $\Sigma_N$  whose eigenvalues decay toward zero. Rather, the probability density concentrates toward the span of eigenvectors of  $\Sigma_N$  whose eigenvalues are bounded away from zero.

The probability density of  $\tilde{\mathbf{r}}'_N$  can be viewed as a function of  $(\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N)^T \Sigma_N^{-1} (\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N)$ . This fact can be used to define  $R_{\alpha, N} \subset \mathbb{R}^{D-1}$ :

$$R_{\alpha, N} = \{ \tilde{\mathbf{r}}' \in \mathbb{R}^{D-1} \mid (\tilde{\mathbf{r}}' - \hat{\mathbf{r}}'_N)^T \Sigma_N^{-1} (\tilde{\mathbf{r}}' - \hat{\mathbf{r}}'_N) \leq \alpha \}. \quad (67)$$

For  $\varepsilon \in (0, 1)$ , define  $\alpha_0(\varepsilon)$  as the maximum possible value of  $\alpha$  such that  $P(\tilde{\mathbf{r}}' \in R_\alpha) \geq 1 - \varepsilon$ :

$$\alpha_0(\varepsilon) = \max_{\text{s.t. } P(\tilde{\mathbf{r}}' \in R_\alpha) \geq 1 - \varepsilon} \alpha. \quad (68)$$

Thus,  $R_{\alpha_0(\varepsilon)}$  contains at least  $1 - \varepsilon$  of the probability density of  $\tilde{\mathbf{r}}'_N$ .

Recall that  $\Sigma_N^{-1}$  has  $r$  eigenvalues that are upper-bounded, while the other  $D - 1 - r$  eigenvalues of  $\Sigma_N^{-1}$  all approach infinity as  $N \rightarrow \infty$ . Let  $(\lambda_i, \boldsymbol{\mu}_i)$  represent the eigenvalues and eigenvectors of  $\Sigma_N^{-1}$ , respectively. Then,  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  can be expressed in terms of the eigenbasis:  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N = \sum_{i=1}^{D-1} \eta_i \boldsymbol{\mu}_i$  for some coefficients  $\eta_i$ , and:

$$(\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N)^T \Sigma^{-1} (\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N) = \sum_{i=1}^{D-1} \eta_i^2 \lambda_i. \quad (69)$$

Then,  $R_{\alpha_0(\varepsilon)}$  can be written as,

$$R_{\alpha_0(\varepsilon)} = \left\{ \tilde{\mathbf{r}}'_N \in \mathbb{R}^{L-1} \left| \sum_{i=1}^{D-1} \eta_i^2 \lambda_i \leq \alpha_0(\varepsilon) \right. \right\}. \quad (70)$$

We can divide  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  for  $\tilde{\mathbf{r}}'_N \in R_{\alpha_0(\varepsilon)}$  into two terms:

$$\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N = \sum_{i=1}^{D-1-r} \eta_i \boldsymbol{\mu}_i + \sum_{i=D-r}^{D-1} \eta_i \boldsymbol{\mu}_i, \quad (71)$$

where  $\lambda_i \rightarrow \infty$  for  $i \leq D - 1 - r$ . The first term contains eigenvectors  $\boldsymbol{\mu}_i$  corresponding to eigenvalues of  $\Sigma_N^{-1}$  that grow to infinity, while the second term contains eigenvectors corresponding to upper-bounded eigenvalues of  $\Sigma_N^{-1}$ . For  $\tilde{\mathbf{r}}'_N$  belonging to  $R_{\alpha_0(\varepsilon)}$ , as  $\lambda_i \rightarrow \infty$ , the  $\eta_i$  coefficients in the first term will decay; otherwise, the sum  $\sum_{i=1}^{D-1} \eta_i^2 \lambda_i$  grows unboundedly, while  $\sum_{i=1}^{D-1} \eta_i^2 \lambda_i$  remains comparatively-smaller for increasingly-many vectors that place weight only on  $\boldsymbol{\mu}_i$  for  $i > D - 1 - r$ . In other words, as  $N \rightarrow \infty$ , the probability density of  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  increasingly concentrates toward vectors that can be expressed as linear combinations of eigenvectors of  $\Sigma_N^{-1}$  corresponding to upper-bounded  $\lambda_i$ . Thus, as long as  $\lambda_j, j > D - 1 - r$ , are upper-bounded, the probability of sampling  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  such that  $\frac{\eta_j}{\sum_{i=1}^{D-1} \eta_i} \geq b$  for any  $j > D - 1 - r$  and any  $b \in (0, 1)$  does not decay to zero.

Next, we argue that the eigenvectors of  $\Sigma_N^{-1}$  approach a set of vectors spanning the null space of  $\hat{M}_2^T \hat{M}_2$  as  $N \rightarrow \infty$ . For  $j \leq D - 1 - r$ ,  $(\lambda_j, \boldsymbol{\mu}_j)$  is an eigenpair of  $\Sigma_N^{-1}$  for which  $\lambda_j \rightarrow \infty$  as  $N \rightarrow \infty$ . The eigenpair relationship gives:

$$\left( \sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T + V_0^{-1} \right) \boldsymbol{\mu}_j = \lambda_j \boldsymbol{\mu}_j. \quad (72)$$

Dividing by  $\lambda_j$ ,

$$\left( \frac{1}{\lambda_j} \sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T + \frac{1}{\lambda_j} V_0^{-1} \right) \boldsymbol{\mu}_j = \boldsymbol{\mu}_j. \quad (73)$$

The term  $\frac{1}{\lambda_j} V_0^{-1}$  becomes insignificant as  $\lambda_j \rightarrow \infty$ , since  $V_0$  is constant, while scaling the first term by  $\frac{1}{\lambda_j}$  does not affect its eigenbasis. Thus, for  $j \leq D - 1 - r$ ,  $\boldsymbol{\mu}_j$  approaches an eigenvector of  $\sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T$  that has an unbounded eigenvalue. Meanwhile, the span of eigenvectors  $\boldsymbol{\mu}_j$  for  $j > D - 1 - r$  approaches the span of eigenvectors of  $\sum_{i=1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T$  with upper-bounded eigenvalues. Equivalently, the span of eigenvectors  $\boldsymbol{\mu}_j$  for  $j > D - 1 - r$  approaches the null space of  $\hat{M}_2^T \hat{M}_2 = \sum_{i=m+1}^N \hat{\alpha}_i^{(N)} \mathbf{z}_i \mathbf{z}_i^T$ .

Recall from Statement 1 that the eigenvectors of  $\hat{M}_2^T \hat{M}_2$  are  $\{\nu_i | i = 1, \dots, D-1\}$ . The  $r$  smallest of these eigenvectors correspond to  $\text{Null}(\hat{M}_2^T \hat{M}_2)$ ; so, for  $i > D-1-r$ ,  $\lambda_i$  are upper-bounded and  $\text{Span}(\nu_i, i > D-1-r)$  approaches  $\text{Span}(\nu_i, i > D-1)$ . The probability of sampling  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N = \sum_{i=1}^{D-1} \beta_i \nu_i$ —with  $\beta_i > \varepsilon$  for  $i < D-1-r$  and fixed  $\varepsilon > 0$ —decays as  $N \rightarrow \infty$ . Thus  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  increasingly coincides with  $\text{Null}(\hat{M}_2^T \hat{M}_2)$ . Meanwhile, for  $j$  such that  $\nu_j \in \text{Null}(\hat{M}_2^T \hat{M}_2)$ , the probability of sampling  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  such that  $\frac{|\beta_j|}{\sum_{i=1}^{D-1} |\beta_i|} \geq b$  for any  $b \in (0, 1)$  does not decay to zero.

We have shown that  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N$  increasingly coincides with  $\text{Null}(\hat{M}_2^T \hat{M}_2)$ . To show that **DPS** is guaranteed to draw samples  $\tilde{\mathbf{r}}'_N$  that coincide with this null space, we must consider the effect of  $\hat{\mathbf{r}}'_N$ . It suffices to show that its magnitude does not increase enormously with  $N$ , since the probability of sampling  $\tilde{\mathbf{r}}'_N - \hat{\mathbf{r}}'_N \in \text{Null}(\hat{M}_2^T \hat{M}_2)$  with an arbitrarily-large magnitude does not decay to zero. Note that the magnitude of  $\hat{\mathbf{r}}'_N$  can be considered upper-bounded, since a reward vector can always be scaled arbitrarily without affecting the resultant policy learned by value iteration.  $\square$

**Lemma 2.2** (Proof of Statement 2 in Lemma 2).

*Proof.* As in Statement 1, express the reward vector  $\tilde{\mathbf{r}}'_N$  sampled from the logistic regression posterior in terms of the eigenbasis of  $\hat{M}_2^T \hat{M}_2$ ,  $\{\nu_i | i = 1, \dots, D-1\}$ :  $\tilde{\mathbf{r}}'_N = \sum_{i=1}^{D-1} \beta_i \nu_i$  for some coefficients  $\beta_i$ . Then, as  $N \rightarrow \infty$ , consider  $\nu_j$  in the null-space of  $\hat{M}_2^T \hat{M}_2$ . Assume that there exists  $b \in (0, 1)$  such that  $\frac{|\beta_j|}{\sum_{i=1}^{D-1} |\beta_i|} \geq b$ . We show that for  $b$  high enough, with probability above zero, the policy will sample a trajectory such that the next sampled observation  $\mathbf{w}_i$  has a nonzero component in  $\nu_j$ .

First, note that if  $\mathbf{z}_i = \sum_{k=1}^{D-1} \gamma_k \nu_k$ , then  $\nu_j^T \mathbf{z}_i = \sum_{k=1}^{D-1} \gamma_k \nu_j^T \nu_k = \sum_{k=1}^{D-1} \gamma_k \delta_{jk} = \gamma_j$ . Thus, because the eigenvectors  $\{\nu_i\}$  form an orthonormal basis, for  $\mathbf{w}_i$  to have a nonzero component  $\gamma_j$  in  $\nu_j$  is equivalent to  $\mathbf{z}_i^T \nu_j \neq 0$ .

By assumption—that is, our ability to choose  $b$  arbitrarily close to 1—we can assume that  $\tilde{\mathbf{r}}'_N$  is arbitrarily aligned with  $\nu_j$ : for any  $\varepsilon_0 > 0$ , we can assume that  $\|\alpha \tilde{\mathbf{r}}'_N - \nu_j\|_2 < \varepsilon_0$ . Define  $\mathbf{z} := -\alpha \tilde{\mathbf{r}}'_N + \nu_j$ . Then,  $\|\mathbf{z}\|_2 < \varepsilon_0$  and  $\nu_j = \alpha \tilde{\mathbf{r}}'_N + \mathbf{z}$ .

We aim to show that  $\mathbf{z}_i^T \nu_j \neq 0$ . We can write,  $\mathbf{z}_i^T \nu_j = \mathbf{z}_i^T (\alpha \tilde{\mathbf{r}}'_N + \mathbf{z}) = \alpha \mathbf{z}_i^T \tilde{\mathbf{r}}'_N + \mathbf{z}_i^T \mathbf{z}$ . The second term has upper-bounded magnitude:

$$|\mathbf{z}_i^T \mathbf{z}| \leq \|\mathbf{z}_i\|_2 \|\mathbf{z}\|_2, \text{ by the Cauchy-Schwarz inequality} \quad (74)$$

$$\leq \|\mathbf{z}_i\|_2 \varepsilon_0 = \varepsilon_0 \sqrt{\mathbf{z}_i^T \mathbf{z}_i} = \varepsilon_0 \sqrt{\mathbf{x}_i^T \mathbf{x}_i}, \text{ because Equation (4) preserves inner products} \quad (75)$$

$$= \varepsilon_0 \|\mathbf{x}_i\|_2 \leq \varepsilon_0 \|\mathbf{x}_i\|_1, \text{ since } \|\mathbf{y}\|_2 \leq \|\mathbf{y}\|_1 \text{ holds for arbitrary vector } \mathbf{y} \in \mathbb{R}^n \quad (76)$$

$$\leq 2\varepsilon_0 h, \text{ where } h \text{ is the episode horizon,} \quad (77)$$

where the final inequality holds because  $\mathbf{x}_i = \mathbf{x}_{i1} - \mathbf{x}_{i2}$ , and the vectors  $\mathbf{x}_{i1}$  and  $\mathbf{x}_{i2}$  contain positive integer elements that sum to  $h$ .

Set  $\varepsilon > 0$ . Since  $|\mathbf{z}_i^T \mathbf{z}| \leq 2\varepsilon_0 h$ , we can set  $\varepsilon_0 \leq \frac{\alpha \varepsilon}{2h}$  to ensure that if  $\|\mathbf{z}\|_2 = \|\alpha \tilde{\mathbf{r}}'_N - \nu_j\|_2 < \varepsilon_0$ , then  $|\mathbf{z}_i^T \mathbf{z}| < \alpha \varepsilon$ . To show that  $\mathbf{z}_i^T \nu_j = \alpha \mathbf{z}_i^T \tilde{\mathbf{r}}'_N + \mathbf{z}_i^T \mathbf{z} \neq 0$ , it thus suffices to show that  $|\mathbf{z}_i^T \tilde{\mathbf{r}}'_N| > \varepsilon$  with nonzero probability, for some  $\varepsilon > 0$  that we specify (note that selecting the value of  $\varepsilon$  determines the possible values of  $b$ ).

Observe that  $|\mathbf{z}_i^T \tilde{\mathbf{r}}'_N| = |\mathbf{x}_i^T \tilde{\mathbf{r}}_N| = |(\mathbf{x}_{i1} - \mathbf{x}_{i2})^T \tilde{\mathbf{r}}_N|$ , again because the linear transformation  $\mathbf{z}_i = V^T \mathbf{x}_i$  (4) preserves inner products. This is the difference in the total rewards of the two trajectories. Recall that the probability of at least a certain component of  $\tilde{\mathbf{r}}'_N$  belonging to  $\text{Null}(\hat{M}_2^T \hat{M}_2)$  is non-decaying. For  $\tilde{\mathbf{r}}'_N \in \text{Null}(\hat{M}_2^T \hat{M}_2)$ , the total reward is zero for any  $\mathbf{z}_i$  in the row-space of  $\hat{M}_2$ . Thus,  $\alpha \tilde{\mathbf{r}}'_N = \nu_j$  maximally encourages the induced policy to explore necessary parts of the state/action space that lead to increasing the row-rank of  $\hat{M}_2$ .

We have  $\|\alpha \tilde{\mathbf{r}}'_N - \nu_j\|_2 < \varepsilon_0$ . The optimal policy is determined by comparing value functions of competing policies; value functions are continuous in rewards, so  $\varepsilon_0$  can be set small enough that this necessary exploration occurs with positive probability.  $\square$

**Lemma 3** (Convergence of dynamics model). *Given Lemma 2, DPS's dynamics model converges to the true dynamics, and all eigenvalues of  $\sum_{i=1}^k f(z_i^T \bar{r}') z_i z_i^T$  approach infinity as  $k \rightarrow \infty$ .*

*Proof.* DPS updates and samples from the dynamics model via the same procedure as the posterior sampling RL algorithm of Osband et al. [29], which conducts posterior sampling RL with numerical rewards; the difference between the two algorithms lies in their handling of rewards, but not of dynamics. In [29], the theoretical result uses the fact that as a particular state/action pair is observed infinitely-often, samples of the dynamics parameters  $p(s'|s, a)$  concentrate to their true values. This fact comes from Lemma 17 in [24], which proves that the deviation between the empirical mean and true value of  $p(s'|s, a)$  decays as the number of visits to state/action pair  $(s, a)$  increases. This lemma is proven via the following concentration inequality:

$$\mathbb{P} \{ \|\hat{p}(\cdot) - p(\cdot)\|_1 \geq \varepsilon \} \leq (2^S - 2) \exp \left( -\frac{n\varepsilon^2}{2} \right), \quad (78)$$

where  $S$  is the size of the state space,  $n$  is the number of times that  $(s, a)$  has been visited,  $p(s, a)$  is the *true* vector of transition probabilities  $[p(s_1|s, a), p(s_2|s, a), \dots, p(s_S|s, a)]^T$ , and  $\hat{p}(s, a)$  is the empirical mean of the observations of  $p(s, a)$ .

We apply the following two results to prove Lemma 3:

1. For any state/action pair that is sampled infinitely-often, the posterior over that state/action's transition probabilities (i.e., the distribution over possible next states) converges to its true values. Thus, if every state/action pair is sampled infinitely-often, then the dynamics model as a whole converges to the true dynamics.
2. Assuming a given dynamics model, the utility model is such that all eigenvalues of  $M^T M$  approach infinity as infinitely-many samples are collected, with  $M$  defined in (62)-(63); Lemma 2 demonstrates that Bayesian logistic regression satisfies this requirement.

The proof of Lemma 3 consists of proving the two statements below:

- A) If the sampled dynamics  $\tilde{p}(s, a)$  are sufficiently-close to the true dynamics (in  $l_1$ -norm  $\|\tilde{p}(s, a) - p(s, a)\|_1$ ) with high probability and  $M_2^T M_2$  is not full-rank, then there exists  $c \in (0, 1)$  such that  $P(\text{Sample } z_i \notin \text{Range}(M_2^T M_2)) \geq c > 0$ .
- B) Every state/action pair will be sampled infinitely-often.

#### **Proof of Statement A:**

Let  $v \in \text{Null}(M_2^T M_2)$ . Treating sampled dynamics model  $\tilde{p}$  as if it were the true dynamics, Lemma 2 establishes that  $P(\text{Sample } z_i \text{ s.t. } z_i^T v \neq 0 | \tilde{p}) \geq f(\tilde{p}) > 0$ , where the lower bound  $f(\tilde{p})$  depends on the sampled dynamics  $\tilde{p}$ , and the probability  $P(\cdot)$  is with respect to sampling the reward model (which determines the policy to execute) and the dynamics transitions during the policy roll-out.

We argue that there exists such a lower bound  $f$  that is *continuous* in the dynamics  $\tilde{p}$ . To do so, we define  $g(\tilde{p}) := P(\text{Sample } z_i \text{ s.t. } z_i^T v \neq 0 | \tilde{p})$ , and show that  $g(\tilde{p})$  has at most finitely-many discontinuities. This is sufficient to show that there must exist a *continuous* function  $f$  such that for any dynamics  $\tilde{p}$ ,  $g(\tilde{p}) \geq f(\tilde{p}) > 0$ .

The rewards are sampled independently of the dynamics, and the probability of sampling rewards that are aligned with  $v$  to at least a specified degree does not decay:  $P \left( \left| \frac{(\bar{r}')^T}{\|\bar{r}'\|_2} \cdot \frac{v}{\|v\|_2} \right| > b \right) \not\rightarrow 0$ , for any  $b \in (0, 1)$ . Statement A considers only a specific iteration of the algorithm, rather than a sequence of episodes, and so the reward distribution can be treated as fixed.

Given dynamics  $\tilde{p}$ , the probability of sampling any trajectory  $\tau = \{s_1, a_1, s_2, a_2, \dots, s_{h+1}\}$  under policy  $\pi$  is continuous in  $\tilde{p}$ :

$$P(\tau) = \tilde{p}_0(s_1) \prod_{i=1}^h \pi(a_i | s_i) \tilde{p}(s_{i+1} | s_i, a_i). \quad (79)$$

Letting  $L$  be the set of trajectories such that  $\mathbf{z}_i^T \mathbf{v} \neq 0$ , the probability of sampling a trajectory  $\tau \in L$  is:

$$P(\tau \in L) = \sum_{\tau' \in L} P(\tau'), \quad (80)$$

where the probability  $P(\tau')$  is given by (79). The function  $g(\tilde{\mathbf{p}}) = P(\tau \in L | \tilde{\mathbf{p}})$  is therefore continuous in the parameters of  $\tilde{\mathbf{p}}$ . The policy  $\pi$ , meanwhile, depends upon the dynamics  $\tilde{\mathbf{p}}$  and sampled rewards  $\tilde{\mathbf{r}}$ , and is selected via value iteration to optimize the expected total reward:

$$\pi^* = \operatorname{argmax}_{\pi} V_{\pi,1}^{\tilde{M}}(s) = \operatorname{argmax}_{\pi} \mathbb{E}_{M,\pi} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right], \quad (81)$$

where  $\tilde{M}$  is the sampled MDP (i.e., consisting of  $\tilde{\mathbf{p}}$  and  $\tilde{\mathbf{r}}$ ).

Value iteration selects the deterministic policy that maximizes the expected total trajectory reward in (81). Given finite state and action spaces and a finite time horizon, there are finitely-many possible deterministic policies. If for given dynamics  $\tilde{\mathbf{p}}$ , only a single policy  $\pi(\tilde{\mathbf{p}})$  maximizes the value function, then under a sufficiently-small change in the dynamics, that same policy  $\pi(\tilde{\mathbf{p}})$  still yields the optimal value function. Under a fixed policy, the probability of generating any trajectory is continuous in the dynamics, and so  $g$  is continuous at  $\tilde{\mathbf{p}}$ .

Thus, the function  $g(\tilde{\mathbf{p}})$  can only be discontinuous at points in the dynamics space at which multiple policies maximize (81). We argue that  $g(\tilde{\mathbf{p}})$  has finitely-many such discontinuities. If this were not the case, there would exist a region of the dynamics space in which the policy maximizing (81) changes infinitely-many times. Since the set of dynamics models is bounded (each dynamics parameter is in  $[0, 1]$ ), this would imply that for any  $\varepsilon > 0$ , there exists a region of the dynamics parameter space with area below  $\varepsilon$ , in which the policy optimizing (81) changes infinitely-many times. To see that this is impossible, notice that any dynamics  $\tilde{\mathbf{p}}$  and policy  $\pi$  induce a distribution  $\tilde{n}(s, a)$  over the expected number of times a trajectory visits each state/action pair. The expected total reward, given  $\tilde{n}(s, a)$  and sampled rewards  $\tilde{\mathbf{r}}$ , can be expressed as,

$$V_{\pi,1}^{\tilde{M}}(s) = \sum_{(s,a) \in S \times \mathcal{A}} \tilde{n}(s, a) \tilde{r}(s, a). \quad (82)$$

Since  $\tilde{n}(s, a)$  depends on the policy and dynamics according to (79) for each possible trajectory, its slope with respect to the dynamics is upper-bounded, and it cannot oscillate infinitely with respect to the dynamics.

Given the continuous lower bound  $f$ , for  $\varepsilon > 0$ , there exists  $\delta$  such that if  $\|\tilde{\mathbf{p}} - \mathbf{p}\|_1 < \delta$ , then  $|f(\tilde{\mathbf{p}}) - f(\mathbf{p})| < \varepsilon$ . If we set  $\varepsilon < \frac{f(\mathbf{p})}{2}$ , then  $P(\text{Sample } \mathbf{z}_i \text{ s.t. } \mathbf{z}_i^T \mathbf{v} \neq 0 | \mathbf{p}) \geq f(\mathbf{p}) > \frac{f(\tilde{\mathbf{p}})}{2} > 0$ .

#### **Proof of Statement B:**

Statement B asserts that every state/action pair will be sampled infinitely-often. To show this, we first define known and unknown state/action pairs:

**Definition 3.**  *$(\varepsilon, \delta)$ -known state/action pair:* Fix  $\varepsilon, \delta > 0$ , and let  $\tilde{\mathbf{p}}(s, a)$  be a sample from the dynamics model's posterior. An  $(\varepsilon, \delta)$ -known state/action pair is one for which  $\|\tilde{\mathbf{p}}(s, a) - \mathbf{p}(s, a)\|_1 < \varepsilon$  with probability  $1 - \delta$ .

**Definition 4.** *Unknown state/action pair:* An  $(\varepsilon, \delta)$ -unknown state/action pair is one that does not satisfy Definition 3.

Previously, we noted that for any state/action pair that is sampled infinitely-often, the model of that state/action's transition probabilities (that is, the distribution over possible next states) converges to the true values. For any fixed  $\varepsilon$  and  $\delta$ , after enough steps in the MDP are taken, at least one state/action pair is guaranteed to eventually be sampled enough that it becomes  $(\varepsilon, \delta)$ -known.

To prove Statement B, it suffices to show that if part of the dynamics are known and part are unknown, then the MDP is guaranteed to eventually leave its known portion. We will prove this by contradiction: we will assume that all sampled policies stay only in the known portion of the MDP, and then contradict this assumption by showing that the MDP is guaranteed to exit the known region.



For  $\varepsilon, \delta > 0$ , an  $(\varepsilon, \delta)$ -known state/action pair is one for which  $\|\tilde{\mathbf{p}}(s, a) - \mathbf{p}(s, a)\|_1 < \varepsilon$  with probability  $1 - \delta$ . For unknown state/action pairs  $(s, a)$ , due to the assumption that only known state/actions are visited, the distribution from which  $\tilde{\mathbf{p}}(s, a)$  is sampled does not change. Without loss of generality, let  $\tilde{s}_1$  be an unknown state/action pair. This means that  $\tilde{s}_1$  is not visited past some iteration  $m$ .

Without visiting  $\tilde{s}_1$ ,  $M_2^T M_2$  cannot be full-rank. As a result, there exists  $\mathbf{v} \in \text{Null}(M_2^T M_2)$  which directly corresponds to the lack of  $\tilde{s}_1$  observations. As in the proof of Statement A, there exists a function  $f$  such that  $P(\text{Sample } \mathbf{z}_i \text{ s.t. } \mathbf{z}_i^T \mathbf{v} \neq 0 | \tilde{\mathbf{p}}) \geq f(\tilde{\mathbf{p}}) > 0$ . While  $\tilde{\mathbf{p}}$  itself changes in each episode, we can show that  $\mathbb{E}[f(\tilde{\mathbf{p}})]$  does not decay to zero. To do so, denote the samples of the known and unknown portions of the dynamics parameters as  $\tilde{\mathbf{p}}_{\text{known}}$  and  $\tilde{\mathbf{p}}_{\text{unknown}}$ , respectively. Also, let  $S_\varepsilon$  be the set of *known* dynamics parameters  $\tilde{\mathbf{p}}_{\text{known}}$  that lie inside the  $\varepsilon_1$ -ball  $\|\tilde{\mathbf{p}}(s, a) - \mathbf{p}(s, a)\|_1 < \varepsilon$  for known  $(s, a)$ . Then:

$$\mathbb{E}[f(\tilde{\mathbf{p}})] = \int \Pr(\tilde{\mathbf{p}}) f(\tilde{\mathbf{p}}) d(\tilde{\mathbf{p}}) \quad (83)$$

$$\geq \min_{\tilde{\mathbf{p}}_{\text{known}} \in S_\varepsilon} \int \Pr(\tilde{\mathbf{p}}_{\text{unknown}}) f(\tilde{\mathbf{p}}) d(\tilde{\mathbf{p}}_{\text{unknown}}) \quad (\text{with prob. } \geq 1 - \delta). \quad (84)$$

By minimizing the known dynamics over the area in which they lie with high probability, we obtain an integral in which all dynamics parameters are either fixed (i.e., the known dynamics) or drawn from an unchanging distribution (i.e., the unknown dynamics). Therefore, since  $f(\tilde{\mathbf{p}})$  is strictly positive everywhere, and we are integrating it over an unchanging probability distribution, the integral must evaluate to a strictly positive, unchanging quantity.

Hence, as long as  $\tilde{s}_1$  is unknown and not visited,  $\mathbb{E}_{\tilde{\mathbf{p}}}[P(\text{Sample } \mathbf{z}_i \text{ s.t. } \mathbf{z}_i^T \mathbf{v} \neq 0)] \geq c > 0$  with probability  $\geq 1 - \delta$ , for some  $c > 0$ . In general, when sampling a sequence of random variables  $X_i$  with bounded support and  $\mathbb{E}[X_i] \geq c$  for all  $i$ , we observe  $X_i \geq c$  infinitely-often. Since  $f(\cdot)$  has support in  $[0, 1]$ , we are guaranteed to infinitely-often sample dynamics  $\tilde{\mathbf{p}}$  such that  $P(\text{Sample } \mathbf{z}_i \text{ s.t. } \mathbf{z}_i^T \mathbf{v} \neq 0) \geq c > 0$ , in which event state/action  $\tilde{s}_1$  has nonzero probability of being sampled. This completes the proof by contradicting the hypothesis that the MDP will only visit known state/action pairs.  $\square$

**Theorem 1** (Asymptotic consistency of DPS). *If there exists a reward function such that a logistic regression model explains the user's preferences, then DPS with a Bayesian logistic regression credit assignment model will learn an asymptotically consistent reward model.*

*Proof.* To prove this result, we apply Proposition 1. Given that Lemma 2, Lemma 3, and Condition 1 hold, it remains only to argue that Condition 2 must hold. To see this, refer back to the proof of Lemma 2. As long as a particular eigenvalue of  $Z^T Z$  does not increase, there is a non-decaying probability of sampling an observation that increases it by at least some minimum amount. The probabilities of sampling such vectors depend on the ratios of the covariance matrix  $\Sigma_k$ 's eigenvalues, and so the ratio of  $\lambda_1$  and  $\lambda_{D-1}$  must always have some upper bound.  $\square$

We turn next to characterizing the regret rate of DPS. We will apply two prior results, one from Gouriou and Monfort [21] regarding the asymptotic distribution of the logistic regression maximum likelihood estimate, while the other (Osband et al. [29]) presents a regret bound for posterior sampling RL.

**Proposition 2** (Asymptotic normality of logistic regression maximum likelihood estimator [21]). *If Conditions 1 and 2 are satisfied, and if  $\hat{\mathbf{r}}'_{ML,k}$  converges almost surely to the true value  $\bar{\mathbf{r}}'$ , then:*

$$\left[ \sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_{ML,k}) \mathbf{z}_i \mathbf{z}_i^T \right]^{\frac{1}{2}} (\hat{\mathbf{r}}'_{ML,k} - \bar{\mathbf{r}}') \xrightarrow{D} \mathcal{N}(\mathbf{0}, \mathbb{I}) \text{ as } k \rightarrow \infty, \quad (85)$$

where  $\xrightarrow{D}$  implies convergence in distribution and  $Q^{\frac{1}{2}}$  is the positive definite matrix associated with positive definite matrix  $Q$  such that  $[Q^{\frac{1}{2}}]^2 = Q$ .

*Proof.* See Proposition 4 in Gouriou and Monfort [21].  $\square$

**Remark:** Just as with Proposition 1, the proof of Proposition 2 in [21] can be adapted such that the result holds when the maximum likelihood estimator  $\hat{\mathbf{r}}'_{ML,k}$  is replaced with the MAP estimator  $\hat{\mathbf{r}}'_k$ .

**Proposition 3** (Expected regret of posterior sampling RL). *In posterior sampling RL with episode horizon  $h$  and numbers of states and actions  $S$  and  $A$ , the expected  $T$ -step regret is bounded as:*

$$\mathbb{E}[\text{Regret}(T, \pi_h^{PS})] = O\left(hS\sqrt{AT\log(SAT)}\right). \quad (86)$$

*Proof.* See Theorem 1 in Osband et al. [29].  $\square$

Next, we show that under preference feedback, the regret can be decomposed into two terms: one that reflects the converging dynamics model, and one that reflects the converging reward model.

**Lemma 4** (Regret decomposition). *The expected regret of DPS can be decomposed into two terms. One of these terms can be upper bounded by the same regret bound as in Osband et al. [29], stated in Proposition (3). The second term may be upper-bounded by*

$$h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{\mathbf{r}} - \tilde{\mathbf{r}}_k\|_\infty] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\hat{\mathbf{r}}_k - \bar{\mathbf{r}}\|_\infty] + h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\hat{\mathbf{r}}_k - \tilde{\mathbf{r}}_k\|_\infty]. \quad (87)$$

*Proof.* Recall that the value of state  $s$  at time-step  $i$  and under policy  $\mu$  and MDP  $M$  is:

$$V_{\mu,i}^M(s) := \mathbb{E}_{M,\mu} \left[ \sum_{j=i}^h \bar{r}^M(s_j, a_j) | s_i = s \right], \quad (88)$$

where  $\bar{r}^M(s, a)$  is the average utility for taking action  $a$  in state  $s$  and MDP  $M$ . Then, the regret from episode  $k$  is defined as:

$$\Delta_k := \sum_{s \in \mathcal{S}} p_0(s) (V_{\mu^*,1}^{M^*} - V_{\mu_k,1}^{M^*}), \quad (89)$$

where  $M^*$  is the true (unknown) MDP,  $\mu^*$  is the optimal policy in  $M^*$ , and  $\mu_k$  is the policy that the algorithm follows in episode  $k$ . The total regret over  $T$  time-steps is then:

$$\text{Regret}(T, \pi) := \sum_{k=1}^{\lceil T/h \rceil} \Delta_k. \quad (90)$$

For a sampled MDP  $M_k$ , drawn from the posterior model of the environment, Osband et al. [29] define:

$$\tilde{\Delta}_k := \sum_{s \in \mathcal{S}} p_0(s) (V_{\mu_k,1}^{M_k} - V_{\mu_k,1}^{M^*}). \quad (91)$$

In Osband et al. [29], the authors prove the following regret equivalence result:

$$\mathbb{E} \left[ \sum_{k=1}^m \Delta_k \right] = \mathbb{E} \left[ \sum_{k=1}^m \tilde{\Delta}_k \right], \quad (92)$$

which they use to derive the posterior sampling RL regret bound restated in Proposition 3.

The true (unknown) MDP  $M^*$  consists of the true dynamics  $P^*$  and average rewards  $\bar{r}(s_j, a_j)$ , while  $M_k$  has a *sampled* dynamics model and *sampled* rewards  $\tilde{r}(s_j, a_j)$ . In addition, we introduce  $M_k^*$ , which pairs the *true* dynamics model  $P^*$  with the *sampled* rewards  $\tilde{r}(s_j, a_j)$ . By adding and

subtracting terms based upon  $M_k^*$ , the terms of  $\tilde{\Delta}_k$  can be decomposed:

$$V_{\mu_k,1}^{M_k}(s) - V_{\mu_k,1}^{M^*}(s) = \mathbb{E}_{M_k, \mu_k} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right] - \mathbb{E}_{M^*, \mu_k} \left[ \sum_{j=1}^h \bar{r}(s_j, a_j) | s_1 = s \right] \quad (93)$$

$$= \mathbb{E}_{M_k, \mu_k} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right] - \mathbb{E}_{M_k^*, \mu_k} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right] \quad (94)$$

$$+ \mathbb{E}_{M_k^*, \mu_k} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right] - \mathbb{E}_{M^*, \mu_k} \left[ \sum_{j=1}^h \bar{r}(s_j, a_j) | s_1 = s \right] \quad (95)$$

$$= \mathbb{E}_{M_k, \mu_k} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right] - \mathbb{E}_{M_k^*, \mu_k} \left[ \sum_{j=1}^h \tilde{r}(s_j, a_j) | s_1 = s \right] \quad (96)$$

$$+ \mathbb{E}_{P^*, \mu_k} \left[ \sum_{j=1}^h (\tilde{r}(s_j, a_j) - \bar{r}(s_j, a_j)) | s_1 = s \right], \quad (97)$$

where in the final term, recall that  $P^*$  is the true transition model.

The regret arising from the first two terms of this expression can be upper-bounded by the same regret bound as in [29], stated in (86) above. This holds because the regret bound of [29] arises from both dynamics and reward models that converge based upon direct observations of each model parameter, i.e., direct observations of both state/action transitions and rewards. In the two terms in (96), however, the rewards are identical among the two terms, while the dynamics are modeled identically to the setup of [29].

It remains to consider the third term of the regret decomposition (97), which is the error due to credit assignment. Let  $\text{REG}_T^R$  be the total component of the  $T$ -step regret arising from this credit assignment error term. Our goal is to upper-bound its expectation:

$$\mathbb{E}[\text{REG}_T^R] = \mathbb{E}_{p_0, \bar{r}} \left\{ \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}_{P^*, \mu_k} \left[ \sum_{j=1}^h (\tilde{r}(s_j, a_j) - \bar{r}(s_j, a_j)) | s_1 = s \right] \right\}. \quad (98)$$

Note that because  $M^*$  and  $M_k^*$  have the same transition dynamics and initial state distributions  $p_0$ , and that because the expectation over both summations in (98) is taken with respect to the same policy  $\mu_k$ , the two MDPs  $M^*$  and  $M_k^*$  have the same probabilities of reaching any state/action pair  $(s, a)$  at each step  $j$ .

Recall that  $\bar{r} \in \mathbb{R}^D$  is the vector of true state/action rewards,  $\tilde{r}_k \in \mathbb{R}^D$  is the vector of sampled state/action rewards (from the model posterior) in episode  $k$ , and  $\hat{r}_k \in \mathbb{R}^D$  is the expected value of the reward model's posterior distribution at episode  $k$ .

Given the observation that  $M^*$  and  $M_k^*$  have the same probabilities of reaching any state/action pair  $(s, a)$  at each step  $j$ , we can write:

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{r} - \tilde{r}_k\|_\infty]. \quad (99)$$

Applying the triangle inequality yields:

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{r} - \tilde{r}_k + \hat{r}_k - \hat{r}_k\|_\infty] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\hat{r}_k - \bar{r}\|_\infty] + h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\hat{r}_k - \tilde{r}_k\|_\infty]. \quad (100)$$

Thus, to bound the total credit assignment error in (98), we must consider the rates at which  $\hat{r}_k$  converges to  $\bar{r}$  and at which  $\tilde{r}_k$  converges to  $\hat{r}_k$ .

□

The final two lemmas characterize the convergence of  $\tilde{\mathbf{r}}_k$  to  $\hat{\mathbf{r}}_k$ , and of  $\hat{\mathbf{r}}_k$  to  $\bar{\mathbf{r}}$ , respectively.

**Lemma 5.** *Sampling  $\tilde{\mathbf{r}}'_k$  via the Laplace approximation to the Bayesian logistic regression posterior gives  $\tilde{\mathbf{r}}'_k \sim \mathcal{N}(\hat{\mathbf{r}}'_k, \Sigma_k)$ , with  $\hat{\mathbf{r}}'_k$  and  $\Sigma_k$  defined in (9) and (10). All eigenvalues of  $\Sigma_k^{-1}$  approach infinity as  $k \rightarrow \infty$ . Moreover, given Condition 2, there exists  $c_0$  such that  $\lambda_{D-1}^{(k)} \geq kc_0$ , where recall that  $\lambda_{D-1}^{(k)}$  is the smallest eigenvalue of  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$ . Thus, asymptotically  $\lambda_{D-1}^{(k)}(\Sigma_k^{-1}) \geq kc_0$ , where  $\lambda_{D-1}^{(k)}(\Sigma_k^{-1})$  is the smallest eigenvalue of  $\Sigma_k^{-1}$ .*

*Proof.* From Theorem 1 (see subsequent remark),  $\hat{\mathbf{r}}'_k \rightarrow \bar{\mathbf{r}}'$  almost surely. Due to this convergence,

$$\Sigma_k^{-1} = \sum_{i=1}^k \frac{\exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)}{(1 + \exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k))^2} \mathbf{z}_i \mathbf{z}_i^T \xrightarrow{k \rightarrow \infty} \sum_{i=1}^k \frac{\exp(\mathbf{z}_i^T \bar{\mathbf{r}}')}{(1 + \exp(\mathbf{z}_i^T \bar{\mathbf{r}}'))^2} \mathbf{z}_i \mathbf{z}_i^T, \quad (101)$$

and all eigenvalues of  $\Sigma_k^{-1}$  approach infinity as  $k \rightarrow \infty$  by Lemma 3. Also, Condition 2 asymptotically holds for  $\Sigma_k^{-1}$  as  $k$  increases, so that if  $\lambda_1^{(k)}(\Sigma_k^{-1})$  and  $\lambda_{D-1}^{(k)}(\Sigma_k^{-1})$  are the largest and smallest eigenvalues of  $\Sigma_k^{-1}$ , respectively, then  $\exists M_1$  such that  $\frac{\lambda_1^{(k)}(\Sigma_k^{-1})}{\lambda_{D-1}^{(k)}(\Sigma_k^{-1})} \leq m_1$  for large enough  $k$ .

To complete the proof, we argue that  $\text{Tr}(\Sigma_k^{-1})$  increases by a lower-bounded amount on average with each time step, where  $\text{Tr}$  denotes the trace, or sum of a matrix's eigenvalues. Combining that 1)  $\frac{\lambda_1^{(k)}(\Sigma_k^{-1})}{\lambda_{D-1}^{(k)}(\Sigma_k^{-1})} \leq m_1$  for large enough  $k$ , and 2)  $\text{Tr}(\Sigma_k^{-1}) = \sum_{i=1}^{D-1} \lambda_i(\Sigma_k^{-1})$  increases (on average) by a lower-bounded amount with each new iteration, one can see that there exists  $c_0$  such that asymptotically  $\lambda_{D-1}^{(k)}(\Sigma_k^{-1}) \geq kc_0$ .

The quantity  $\text{Tr}(\Sigma_k^{-1})$  can only increase:

$$\text{Tr}[\Sigma_k^{-1}] = \sum_{i=1}^k \frac{\exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)}{(1 + \exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k))^2} \text{Tr}(\mathbf{z}_i \mathbf{z}_i^T), \text{ by linearity of trace} \quad (102)$$

$$= \sum_{i=1}^k \frac{\exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)}{(1 + \exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k))^2} \mathbf{z}_i^T \mathbf{z}_i, \text{ by the cyclic property of trace} \quad (103)$$

$$= \sum_{i=1}^k \frac{\exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)}{(1 + \exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k))^2} \mathbf{x}_i^T \mathbf{x}_i, \text{ by (6).} \quad (104)$$

The quantity  $\frac{\exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k)}{(1 + \exp(\mathbf{z}_i^T \hat{\mathbf{r}}'_k))^2}$  is bounded away from zero because it approaches  $\frac{\exp(\mathbf{z}_i^T \bar{\mathbf{r}}')}{(1 + \exp(\mathbf{z}_i^T \bar{\mathbf{r}}'))^2}$ , which is bounded away from zero as justified previously. For any observation  $\mathbf{x}_i$  such that  $\mathbf{x}_i = \mathbf{x}_{i1} - \mathbf{x}_{i2} \neq 0$ ,  $\mathbf{x}_i^T \mathbf{x}_i \geq 1$  always, since  $\mathbf{x}_{i1}$  and  $\mathbf{x}_{i2}$  differ by at least one. It remains to argue that the event  $\{\mathbf{x}_{i1} - \mathbf{x}_{i2} = 0\}$  cannot occur increasingly-often: there must be some below-one upper-bound to the probability that this event takes place. This is indeed the case; if it were not, this would imply that the model converges toward always sampling a single trajectory  $\mathbf{x}$ ; however, as long as a particular eigenvalue of  $Z^T Z$  does not increase, there is a non-decaying probability of sampling an observation  $\mathbf{x}'$  that increases it by at least some minimum amount (Lemma 1). The probability of sampling a trajectory  $\mathbf{x}' \neq \mathbf{x}$  depends on the ratios of the covariance matrix's eigenvalues, which are bounded according to Condition 2.

□

**Lemma 6.** *Under the conditions for asymptotic consistency of logistic regression (Theorem 1), the vector  $\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  behaves asymptotically as  $\mathcal{N}\left(\mathbf{0}, \left[\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T\right]^{-1}\right)$ . This is the same distribution as that characterizing  $\tilde{\mathbf{r}}'_k - \hat{\mathbf{r}}'_k$  as  $k \rightarrow \infty$ , as discussed in Lemma 5. Thus, similarly,  $\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  asymptotically has covariance  $\Sigma_k$ .*

*Proof.* By Proposition 2 (see subsequent remark),

$$\left[ \sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k) \mathbf{z}_i \mathbf{z}_i^T \right]^{\frac{1}{2}} (\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}') \xrightarrow{D} \mathcal{N}(\mathbf{0}, \mathbb{I}) \text{ as } k \rightarrow \infty. \quad (105)$$

Multiplying both sides by  $\left[ \sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T \right]^{-1/2}$  gives that:

$$\left[ \sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T \right]^{-1/2} \left[ \sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k) \mathbf{z}_i \mathbf{z}_i^T \right]^{1/2} (\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}') \quad (106)$$

$$\text{behaves asymptotically as } \mathcal{N} \left( \mathbf{0}, \left[ \sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T \right]^{-1} \right). \quad (107)$$

Note that  $\left[ \sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T \right]^{-1/2} \left[ \sum_{i=1}^k f(\mathbf{z}_i^T \hat{\mathbf{r}}'_k) \mathbf{z}_i \mathbf{z}_i^T \right]^{1/2} \rightarrow \mathbb{I}$  because  $\hat{\mathbf{r}}'_k \rightarrow \bar{\mathbf{r}}'$ , so that  $\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  behaves asymptotically as  $\mathcal{N} \left( \mathbf{0}, \left[ \sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T \right]^{-1} \right)$ .  $\square$

**Theorem 2** (Asymptotic regret rate of DPS). *If there exists a reward function such that a logistic regression model explains the preferences, then DPS has an asymptotic no-regret rate of  $O \left( hS \sqrt{AT \log(SAT)} + h \sqrt{\frac{SA}{c_0} T \log(T)} \right)$ , where  $c_0$  is a minimum rate at which eigenvalues of  $\sum_{i=1}^k f(\mathbf{z}_i^T \bar{\mathbf{r}}') \mathbf{z}_i \mathbf{z}_i^T$  increase linearly with collection of samples  $\mathbf{z}_i$  that impact those eigenvalues.*

*Proof.* From Lemma 3, the regret is upper-bounded by a sum of two terms, where the first of those terms is bounded by  $O \left( hS \sqrt{AT \log(SAT)} \right)$  as proven in [29]. Here, we analyze the second term, given in (98), and upper-bounded by the expression in (99), restated here:

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{\mathbf{r}} - \tilde{\mathbf{r}}_k\|_\infty].$$

We use that  $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2$  for any  $\mathbf{x} \in \mathbb{R}^n$ , and that the linear transformation in (4) preserves inner products, as established in Equation (6). In particular, the linear transformation preserves  $l_2$ -norms of vectors, since for any vector  $\mathbf{x} \in \mathbb{R}^n$ ,  $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T \mathbf{x}}$ . Applying both of these facts gives:

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{\mathbf{r}} - \tilde{\mathbf{r}}_k\|_\infty] \leq h \sum_{k=1}^{\lceil T/h \rceil} \mathbb{E}[\|\bar{\mathbf{r}}' - \tilde{\mathbf{r}}'_k\|_2]. \quad (108)$$

Consider the expected  $l_2$ -norm of a Gaussian vector  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ . It can be upper-bounded in terms of  $n$  and the eigenvalues of  $\Sigma$ :

$$\mathbb{E}[\|\mathbf{x}\|_2] \leq \sqrt{\mathbb{E}[\|\mathbf{x}\|_2^2]}, \text{ by Jensen's inequality} \quad (109)$$

$$= \sqrt{\mathbb{E} \left[ \sum_{i=1}^n x_i^2 \right]} = \sqrt{\sum_{i=1}^n \mathbb{E}[x_i^2]} = \sqrt{\sum_{i=1}^n \text{Var}[x_i]} \quad (110)$$

$$= \sqrt{\text{Tr}(\Sigma)} = \sqrt{\sum_{i=1}^n \lambda_i(\Sigma)} \leq \sqrt{n \lambda_{\max}(\Sigma)}. \quad (111)$$

Next, consider the asymptotic probability distribution of  $\tilde{\mathbf{r}}'_k - \bar{\mathbf{r}}' = (\tilde{\mathbf{r}}'_k - \hat{\mathbf{r}}'_k) + (\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}')$ . Given  $\bar{\mathbf{r}}'$ , which is constant, the two quantities in parentheses are independent due to the nature of Laplace sampling, in which  $\tilde{\mathbf{r}}'_k$  is drawn from a normal distribution centered at  $\hat{\mathbf{r}}'_k$ . By

Lemmas 5 and 6, both  $\tilde{\mathbf{r}}'_k - \hat{\mathbf{r}}'_k$  and  $\hat{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  behave asymptotically according to  $\mathcal{N}(\mathbf{0}, \Sigma_k)$ , where  $\Sigma_k := \left[ \sum_{i=1}^k f(z_i^T \bar{\mathbf{r}}') z_i z_i^T \right]^{-1}$ . Therefore,  $\tilde{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  behaves asymptotically according to  $\mathcal{N}(\mathbf{0}, 2\Sigma_k)$ .

Combining (108) and (111) gives:

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \sqrt{(D-1)\lambda_{\max}(2\Sigma_k)} \leq h\sqrt{2SA} \sum_{k=1}^{\lceil T/h \rceil} \lambda_{\max}(\Sigma_k). \quad (112)$$

Although  $\lambda_{\max}(\Sigma_k)$  decays to zero over time by Lemma 2, this result is too loose: there could be many consecutive iterations in which  $\lambda_{\max}(\Sigma_k)$  remains constant. For instance, if Condition 2 is enforced explicitly as discussed earlier, then as the model posterior converges, one would expect to find increasingly-many consecutive iterations in which the reward posterior is not updated; in these iterations,  $\lambda_{\max}(\Sigma_k)$  is fixed.

The key intuition, formalized below, is that when a non-optimal episode is sampled, the covariance matrix  $\Sigma_k$  shrinks with respect to the directions (i.e., its eigenvectors) responsible for estimating the rewards suboptimally. Meanwhile, if particular eigenvectors of  $\Sigma_k$  *do not* result in a suboptimal policy during episode  $k$ , then for that episode, the corresponding eigenvalues can be removed from the right-hand sum in (112). Thus, eigenvalues do not appear in the regret's upper bound *except* for when their corresponding eigenvectors affect an episode's regret. In this event, however, the sampled observations cause  $\Sigma_k$  to shrink along the dimensions of the guilty eigenvectors.

To see this, first note that in the context of preference-based learning, the true reward function  $\bar{\mathbf{r}}$  is unobserved. One can define an equivalence class of reward functions that, when coupled with the true transition dynamics, induce the optimal policy with respect to the user's preferences. In the preference-based setting, an optimal policy is defined as any policy  $\pi^*$  such that when compared to any other policy  $\pi_i$ :

$$\mathbb{E}_{\tau^* \sim \pi^*, \tau_i \sim \pi_i} [P(\tau^* > \tau_i)] \geq \frac{1}{2}, \quad (113)$$

where  $\tau^*$  is a trajectory sampled from policy  $\pi^*$ , and  $\tau_i$  is a trajectory sampled from  $\pi_i$ .

Let  $S_{\text{eq}} \subset \mathbb{R}^{D-1}$  be the equivalence class of reward functions that induce policies satisfying (113) when coupled with the true dynamics. To see that there are indeed multiple such functions, notice that for any reward vector  $\mathbf{r}' \in S_{\text{eq}}$ ,  $a\mathbf{r}' + b\mathbf{1} \in S_{\text{eq}}$  for any  $a > 0$  and  $b \in \mathbb{R}$ , where  $\mathbf{1} \in \mathbb{R}^{D-1}$  is a vector of ones. Therefore, in (108), we can minimize each episode's regret over reward vectors belonging to  $S_{\text{eq}}$ :

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \min_{\mathbf{r}' \in S_{\text{eq}}} \left\{ \mathbb{E} \left[ \|\tilde{\mathbf{r}}'_k - \mathbf{r}'\|_2 \right] \right\}. \quad (114)$$

Even given specified scaling and normalization of the rewards (e.g. specified values of the smallest and largest state/action rewards), there could be multiple reward functions that give rise to the optimal policy. For instance, a small perturbation to the rewards might not change the policy. Given specified scaling and normalization, however, there is only one true latent reward function governing the preferences; this is the specific  $\bar{\mathbf{r}}' \in S_{\text{eq}}$  to which the logistic regression posterior converges.

Define  $\Sigma_k := \left[ \sum_{i=1}^k f(z_i^T \bar{\mathbf{r}}') z_i z_i^T \right]^{-1}$ , and let  $\{\mathbf{v}_{k1}, \dots, \mathbf{v}_{k(D-1)}\}$  be an orthonormal basis of eigenvectors of  $\Sigma_k$ . For  $\mathbf{r}' \in S_{\text{eq}}$ , the quantity  $\tilde{\mathbf{r}}'_k - \mathbf{r}'$  can be expressed in this eigenbasis:

$$\mathbb{E}[\text{REG}_T^R] \leq h \sum_{k=1}^{\lceil T/h \rceil} \min_{\mathbf{r}' \in S_{\text{eq}}} \left\{ \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \right] \right\}. \quad (115)$$

Consider the inner sum,  $\sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki}$ . We will show that if any terms in this sum are known *not* contribute to the regret (i.e., to sampling a non-optimal policy in episode  $k$ ), then they can be eliminated from the sum. In particular, if an episode's sampled reward belongs to  $S_{\text{eq}}$ , then its reward regret  $\text{REG}_T^R$  is zero. Otherwise, the components of the reward function that make the policy *non-optimal* cause the corresponding eigenvalues of  $\Sigma_k$  to shrink.

Assume that in episode  $k$ , the reward sample's projection onto  $\mathbf{v}_i$  does not affect the regret. In other words, there exists  $\mathbf{r}' \in S_{\text{eq}}$  such that  $((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} = \mathbf{0}$ . Let  $\mathcal{I}_k \subset \{1, \dots, D-1\}$  be a set of indices that *jointly* do not affect the regret, such that there exists  $\mathbf{r}_k^{*'} \in S_{\text{eq}}$  satisfying:

$$\sum_{i \in \mathcal{I}_k} (\mathbf{v}_i^T (\tilde{\mathbf{r}}'_k - \mathbf{r}_k^{*'})) \mathbf{v}_i = \mathbf{0}. \quad (116)$$

Let  $\mathcal{J}_k$  be the set of indices *not* contained in  $\mathcal{I}_k$ :  $\mathcal{J}_k = \{1, \dots, D-1\} \setminus \mathcal{I}_k$ ; we make no particular assumptions on how vectors in  $\mathcal{J}_k$  affect the regret. The vector  $\mathbf{r}_k^{*'}$  can then be written as:

$$\mathbf{r}_k^{*'} = \sum_{i \in \mathcal{I}_k} (\mathbf{v}_i^T \tilde{\mathbf{r}}'_k) \mathbf{v}_i + \sum_{j \in \mathcal{J}_k} \alpha_j^* \mathbf{v}_j, \quad (117)$$

for some constants  $\alpha_j^*, j \in \mathcal{J}_k$ .

Due to orthogonality of  $\mathbf{v}_i$  and  $\mathbf{v}_j$ , (116) can be modified by adding any linear combination of the vectors  $\mathbf{v}_j, j \in \mathcal{J}_k$ , to  $\mathbf{r}_k^{*'}$ :

$$\sum_{i \in \mathcal{I}_k} \left( \mathbf{v}_i^T \left( \tilde{\mathbf{r}}'_k - \mathbf{r}_k^{*'} - \sum_{j \in \mathcal{J}_k} \alpha_j \mathbf{v}_j \right) \right) \mathbf{v}_i = \mathbf{0}, \text{ for any values of } \alpha_j. \quad (118)$$

Conditioning on knowledge of the set  $\mathcal{I}_k$ , we can upper-bound the minimization in (115) as follows:

$$\min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \quad (119)$$

$$= \min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i \in \mathcal{I}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} + \sum_{j \in \mathcal{J}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \quad (120)$$

$$\leq \min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i \in \mathcal{I}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 + \left\| \sum_{j \in \mathcal{J}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right], \quad (121)$$

where the last step comes from the triangle inequality. The expression can be further upper-bounded by restricting the set over which the minimization takes place:

$$\begin{aligned} & \min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \\ & \leq \min_{\substack{\mathbf{r}' \in S_{\text{eq}} \\ \mathbf{r}' = \mathbf{r}_k^{*'} + \sum_{j \in \mathcal{J}_k} \alpha_j \mathbf{v}_j, \\ \alpha_j \in \mathbb{R}}} \mathbb{E} \left[ \left\| \sum_{i \in \mathcal{I}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 + \left\| \sum_{j \in \mathcal{J}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right]. \end{aligned}$$

Note that the constraint set of the minimization is nonempty, since  $\mathbf{r}_k^{*'} \in S_{\text{eq}}$  and satisfies  $\mathbf{r}' = \mathbf{r}_k^{*'} + \sum_{j \in \mathcal{J}_k} \alpha_j \mathbf{v}_j$  when  $\alpha_j = \alpha_j^* \forall j \in \mathcal{J}_k$ . Next, for any  $\mathbf{r}'$  satisfying the minimization constraints, the first  $l_2$ -norm expression is equal to zero by (118). The second term, meanwhile, is not affected by the projections of  $\mathbf{r}'$  onto  $\mathbf{v}_i$  for any  $i \in \mathcal{I}_k$ , and so the constraint  $\mathbf{r}' = \mathbf{r}_k^{*'} + \sum_{j \in \mathcal{J}_k} \alpha_j \mathbf{v}_j$  is irrelevant to the minimization. This gives:

$$\min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \leq \min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{j \in \mathcal{J}_k} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right].$$

This can again be upper-bounded by setting  $\mathbf{r}'$  to any specific value in  $S_{\text{eq}}$ . In particular, we can set  $\mathbf{r}' = \bar{\mathbf{r}}'$ , the true latent reward function governing the preferences. Recall that while multiple reward

functions may give rise to the optimal policy, the true rewards  $\bar{\mathbf{r}}'$  are unique under fixed scaling and normalization. Therefore,

$$\min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \leq \mathbb{E} \left[ \left\| \sum_{j \in \mathcal{J}_k} ((\tilde{\mathbf{r}}'_k - \bar{\mathbf{r}}')^T \mathbf{v}_{kj}) \mathbf{v}_{kj} \right\|_2 \middle| \mathcal{I}_k \right]. \quad (122)$$

Recall from above that  $\tilde{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  behaves asymptotically as  $\mathcal{N}(\mathbf{0}, 2\Sigma_k)$ . The right-hand-side summation in (122) is the projection of  $\tilde{\mathbf{r}}'_k - \bar{\mathbf{r}}'$  onto the subspace of eigenvectors of  $\Sigma_k$  corresponding to  $\mathcal{J}_k$ . Applying the result in (111) to (122), we finally obtain:

$$\min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \leq \sqrt{\sum_{j \in \mathcal{J}_k} \lambda_j(2\Sigma_k)} \quad (123)$$

$$\leq \sqrt{2SA \max_{j \in \mathcal{J}_k} \lambda_j(\Sigma_k)}. \quad (124)$$

This result demonstrates that if an eigenvector of  $\Sigma_k$  does not affect the regret, then its corresponding eigenvalue can be removed from the regret's upper bound in (111). Therefore, an eigenvector of  $\Sigma_k$  does not affect the regret *unless* its contribution to the sampled reward  $\tilde{\mathbf{r}}_k$  makes the policy non-optimal.

A non-optimal policy's distribution over sampled observations differs from that of the optimal policy. When a non-optimal policy is sampled, the covariance matrix is more likely to shrink along directions of worse observations, decreasing the future probability of sampling a non-optimal policy. This shrinkage occurs linearly with respect to the episodes in which non-optimal policies are sampled because  $(\Sigma_k)^{-1}$  grows linearly in directions favored by the non-optimal policies, with respect to the number of times that they are executed. Indeed, for an arbitrary vector  $\mathbf{v} \in \mathbb{R}^{D-1}$ :

$$\mathbf{v}^T (\Sigma_k)^{-1} \mathbf{v} = \sum_{i=1}^k \alpha_i \mathbf{v}^T (\mathbf{z}_i \mathbf{z}_i^T) \mathbf{v} = \sum_{i=1}^k \alpha_i (\mathbf{z}_i^T \mathbf{v})^2, \quad (125)$$

where  $\alpha_i = f(\mathbf{z}_i^T \bar{\mathbf{r}}')$  lives in a bounded range as discussed in Lemma 1; this quantity increases linearly on average with respect to observations  $\mathbf{z}_i$  that are similarly-aligned with  $\mathbf{v}$ . Also,

$$\mathbf{v}^T (\Sigma_k)^{-1} \mathbf{v} = \mathbf{v}^T \left( \sum_{i=1}^{D-1} [\lambda_i(\Sigma_k)]^{-1} \mathbf{v}_i \mathbf{v}_i^T \right) \mathbf{v} = \sum_{i=1}^{D-1} [\lambda_i(\Sigma_k)]^{-1} (\mathbf{v}_i^T \mathbf{v})^2. \quad (126)$$

Comparing (125) and (126), one can see that the eigenvalues of  $(\Sigma_k)^{-1}$  associated with non-optimal policies should increase linearly with the collection of observations  $\mathbf{z}_i$  associated with those non-optimal policies.

The regret for an episode is zero if the optimal policy is selected. Otherwise, examine the bound from (123):  $\sqrt{2SA \max_{j \in \mathcal{J}_k} \lambda_j(\Sigma_k)} = \sqrt{\frac{2SA}{\min_{j \in \mathcal{J}_k} \lambda_j(\Sigma_k^{-1})}}$ . The denominator increases linearly with respect to episodes with non-optimal policies, so that asymptotically:

$$\min_{\mathbf{r}' \in S_{\text{eq}}} \mathbb{E} \left[ \left\| \sum_{i=1}^{D-1} ((\tilde{\mathbf{r}}'_k - \mathbf{r}')^T \mathbf{v}_{ki}) \mathbf{v}_{ki} \right\|_2 \middle| \mathcal{I}_k \right] \leq \sqrt{\frac{2SA}{c_0 k}}, \quad (127)$$

where  $c_0$  is a constant related to the rate at which eigenvalues of  $(\Sigma_k)^{-1}$  increase with respect to trajectories sampled from non-optimal policies. Summing over all the episodes gives:

$$\mathbb{E}[\text{REG}_T^R] \leq \frac{h\sqrt{2SA}}{c_0} \sum_{k=1}^T \frac{1}{\sqrt{k}} \leq \frac{h\sqrt{2SA}}{c_0} \sqrt{T \log(T)} \text{ for all } T \geq 17. \quad (128)$$

Combining (128) with the regret decomposition result of Lemma 4 and the posterior sampling regret bound from Osband et al. [29] (Proposition 3) yields the stated result.  $\square$



### A2.1 Extending proof techniques to other credit assignment models

Currently, this proof methodology extends only to the Bayesian logistic regression credit assignment model. Therefore, extending it to other credit assignment models, such as the Gaussian process regression and Bayesian linear regression methods detailed in Appendix A1, is an important direction for future work. The concept of regret decomposition, as introduced in Lemma 3, is not dependent on a specific credit assignment model. Thus, the concept of decomposing the total regret into two terms, dependent on the convergence of the dynamics and reward models, respectively, holds under any credit assignment model. For the dynamics-dependent term, the regret result from Osband et al. [29] can be applied. The second, reward-dependent term can be bounded if: (1) posterior samples from the reward model concentrate to the posterior distribution’s MAP estimate, (2) the reward model’s posterior concentrates to the true underlying utilities, and (3) the events in (1) and (2) occur at a sufficiently-fast rate.

We hypothesize that existing results on asymptotic consistency of Bayesian linear regression [19] and Gaussian process regression [34] could be leveraged toward extending notions of consistency and regret toward these credit assignment models. Unlike with classification-based credit assignment, it may be necessary to consider the residuals between preference labels and utilities. The concept of approximate linearity [41] has facilitated bridging the gap between the preference and absolute-reward domains in the bandit setting, and could potentially also apply here. In practice, we expect that DPS would perform well with any asymptotically consistent model for rewards that sufficiently captures the users’ preference behavior.

## A3 Additional experimental details

As described in Section 6, experiments were conducted with the RiverSwim and random MDP environments. We use an episode horizon time of 50 in both cases. Figures 2 and 3 display performance in both environments for four values of the hyperparameter  $c$  ( $c \in \{0.1, 0.5, 1, 1000\}$ ), which governs the degree of preference noise. Experiments were run on an Ubuntu 16.04.3 machine with 16 GB of RAM and an 8-core Intel i7 processor.

We detail the ranges of hyperparameter values tested, as well as those displayed on the plots, for the different algorithms. For DPS, hyperparameters were tuned manually. For Gaussian process regression credit assignment, we used RBF kernels for the Gaussian process model, considering kernel variances from 0.001 to 0.5, kernel lengthscales from 0 to 0.1, and noise variances from 0.001 to 0.1. The plots depict values of (0.03, 0, 0.05), respectively, for these parameter values in the RiverSwim case, while depicting (1, 0, 0.03) in the random MDP case.

For Bayesian linear regression, we considered ranges of hyperparameter values between 0.1 and 3 for both hyperparameters ( $\sigma$  and  $\lambda$ ). The RiverSwim plots display hyperparameter values of 0.5 and 0.1 for  $\sigma$  and  $\lambda$ , respectively, while for the random MDP environment, both are set to 0.1. For Bayesian logistic regression, we set the prior mean to zero for all states and actions. All prior covariance matrices considered were of the form  $b\mathbb{I} \in \mathbb{R}^{SA \times SA}$ , where  $\mathbb{I}$  is the identity matrix, and tested values of  $b$  ranged from 0.1 to 30; the plots all show results for a prior covariance matrix of  $20\mathbb{I}$ .

The EMPC algorithm [50] has two hyperparameter values,  $\alpha$  and  $\eta$ . We optimize both of these jointly via a grid search over values of (0.1, 0.2, ..., 0.9), with 100 runs of each pair of values. The best-performing hyperparameter values (i.e. those achieving the highest total reward) are displayed in Table 1; these are the hyperparameter values depicted in Figures 2 and 3.

Noise:	0.1	0.5	1	1000
RiverSwim	0.1/0.5	0.1/0.7	0.2/0.2	0.1/0.6
Random MDP	0.3/0.8	0.9/0.7	0.4/0.2	0.7/0.4

Table 1: Each table element shows best-performing  $\alpha/\eta$  values for the corresponding simulation domain and noise parameter.

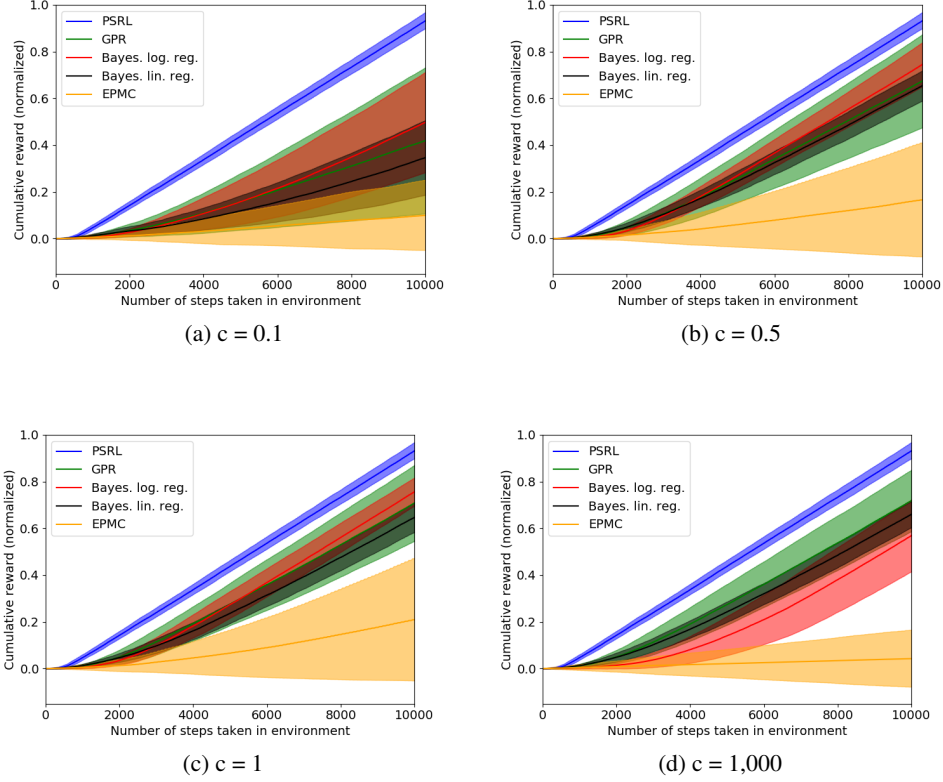


Figure 2: Empirical performance of DPS in the RiverSwim environment for varying values of the noise hyperparameter  $c$ . For trajectories  $\tau_i$  and  $\tau_j$ ,  $P(\tau_i > \tau_j) = \{1 + \exp[-c(\bar{r}(\tau_i) - \bar{r}(\tau_j))]\}^{-1}$ , where  $\bar{r}(\tau_i)$  and  $\bar{r}(\tau_j)$  are the total rewards accrued by the two trajectories. Posterior sampling RL (PSRL) [29] is an upper bound that receives numerical rewards; Gaussian process regression (GPR), Bayesian linear regression, and Bayesian logistic regression are all instances of DPS. EPMC is a baseline from [50] as discussed in Section 6. Normalization is with respect to the total reward achieved by the optimal policy. Plots display the mean  $\pm$  one std over 100 runs of each algorithm tested. Overall, we see that DPS performs well and is robust to the choice of credit assignment model.

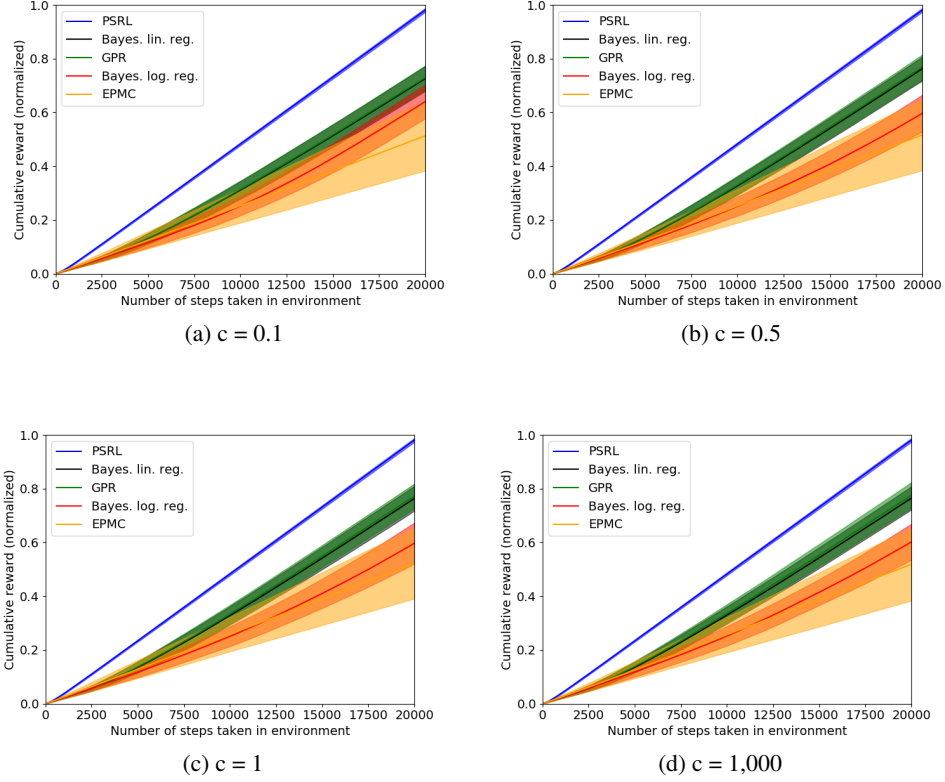


Figure 3: Empirical performance of DPS in the random MDP environment for varying values of the noise hyperparameter  $c$ . For trajectories  $\tau_i$  and  $\tau_j$ ,  $P(\tau_i > \tau_j) = \{1 + \exp[-c(\bar{r}(\tau_i) - \bar{r}(\tau_j))]\}^{-1}$ , where  $\bar{r}(\tau_i)$  and  $\bar{r}(\tau_j)$  are the total rewards accrued by the two trajectories. Posterior sampling RL (PSRL) [29] is an upper bound that receives numerical rewards; Gaussian process regression (GPR), Bayesian linear regression, and Bayesian logistic regression are all instances of DPS. EPMC is a baseline from [50] as discussed in Section 6. Normalization is with respect to the total reward achieved by the optimal policy. Plots display the mean  $\pm$  one std over 100 runs of each algorithm tested. Overall, we see that DPS performs well and is robust to the choice of credit assignment model.